



Universidad Pública de Navarra  
Nafarroako Unibertsitate Publikoa

**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA AGRÓNOMICA Y  
BIOCIENCIAS**

**NEKAZARITZAKO INGENIARITZAKO ETA BIOZIENTZIETAKO GOI  
MAILAKO ESKOLA TEKNIKOA**

*EVALUACIÓN DE VARIABLES LIDAR DEL AÑO 2017 PARA LA GENERACIÓN DE INVENTARIOS DE  
RECURSOS DE BIOMASA FORESTAL DE NAVARRA*

---

*NAFARROAKO BASO-BIOMASAKO BALIABIDEEN INBENTARIOAK SORTZEKO 2017KO LIDAR  
ALDAGAIEN EBALUAZIOA.*

presentado por

*XABIER GOLDARAZ UGARTE(e)k  
aurkeztua*

MÁSTER UNIVERSITARIO EN SISTEMAS DE INFORMACIÓN GEOGRÁFICA Y TELEDETECCIÓN  
UNIBERTSITATE MASTERRA INFORMAZIO GEOGRAFIKOKO SISTEMETAN ETA TELEDETEKZIOAN



septiembre de 2022, 2022 iraila

## RESUMEN

El objetivo de este trabajo ha sido generar modelos de regresión para estimar, a partir de datos LiDAR, la altura dominante, área basimétrica y volumen total en masas de dos especies forestales: haya y pino silvestre. Se ha procesado el vuelo LiDAR adquirido en 2017 sobre Navarra y se han calculado 27 métricas LiDAR de potencial interés forestal. Se ha contado con datos de campo de inventarios forestales para el entrenamiento y validación. Se han ajustado modelos de regresión Random Forests basados en estas métricas LiDAR que se han combinado también con datos de imágenes satelitales Sentinel-1 y Sentinel-2. Los resultados obtenidos han sido satisfactorios, aunque el ajuste de los modelos para la variable volumen total ha sido peor que para las otras dos. En general los modelos LiDAR han proporcionado los mejores ajustes sin que los modelos combinados con Sentinel aporten una mejora clara.

**Palabras clave:** LiDAR, variables forestales, Random Forests, Sentinel-1, Sentinel-2.

## ABSTRACT

The aim of this study is to generate regression models to estimate, from LiDAR data, the dominant height, basal area and total volume in stands of two forest species: beech and scots pine. The LiDAR flight acquired in 2017 over Navarra has been processed and have been calculated 27 LiDAR metrics that are potentially interesting for forestry. Forest inventory field data has been available for training and validation. Random Forests regression models have been fitted based on these LiDAR metrics, which have also been combined with data from Sentinel-1 and Sentinel-2 satellite images. The results obtained have been satisfactory, although the fit of the models for the total volume variable has been worse than for the other two. In general, the LiDAR models have provided the best fits, however, the models combined with Sentinel do not provide a clear improvement.

**Key words:** LiDAR, forest variables, Random Forests, Sentinel-1, Sentinel-2.

## LABURPENA

Lan honen helburua erregresio-ereduak sortzea izan da, LiDAR datuetan oinarrituta, garai nagusia, eremu basimetricoa eta bolumena kalkulatzeko bi baso espezierendako (pagoa eta pinua). 2017an Nafarroan zehar eginiko LiDAR hegaldia prozesatu da, eta baso-interes potentzialeko 27 LiDAR metrika kalkulatu dira. Baso-inbentarioen landa-datuak izan ditu entrenamendurako eta baliozkotzeko. Random Forests erregresio-ereduak doitu dira, LiDAR metrika horietan oinarrituz, eta Sentinel-1 eta Sentinel-2 satellite-irudien datuekin ere konbinatu dira. Lortutako emaitzak onak izan dira, nahiz eta bolumen aldagai modeloen doikuntza beste bietarako baino okerragoa izan den. Oro har, LiDAR ereduak doikuntzarik onenak eman dituzte, Sentinelekin konbinatutako ereduak hobekuntza argirik ekarri gabe.

**Hitz gakoak:** LiDAR, baso-aldagaiak, Random Forests, Sentinel-1, Sentinel-2

---

## AGRADECIMIENTOS

*Quiero agradecer en primer lugar a mi tutor Jesús Álvarez por darme la oportunidad de trabajar con él, y así poder seguir aprendiendo y disfrutando de este gran ámbito que es la Teledetección. También quiero agradecer a mi compañera y amiga Itsaso Aranguren, por su ayuda, por su buena voluntad y, sobre todo, por su paciencia. No me quiero olvidar de mis compañeros del máster de MUSIGT, a los que os doy las gracias por haber estado ahí. Pero en especial quería agradecer a María y a Txaro, por haber comenzado un camino juntos hace dos años y haber estado codo con codo ayudándonos. De corazón, muchísimas gracias.*

## ÍNDICE

<b>1. INTRODUCCIÓN</b>	<b>10</b>
<b>2. OBJETIVOS</b>	<b>13</b>
2.1. OBJETIVOS SECUNDARIOS	13
<b>3. MATERIAL Y MÉTODOS</b>	<b>14</b>
3.1. ÁREA DE ESTUDIO	14
3.2. FLUJO DE TRABAJO	14
3.3. MATERIALES (INICIO)	16
3.3.1. DESCRIPCIÓN DE DATOS DE ORDENACIONES FORESTALES	16
3.3.2. DESCRIPCIÓN DE DATOS LIDAR	20
3.4. METODOLOGÍA (PROCESO)	23
3.4.1. DESCARGA Y PRETRATAMIENTO DE DATOS LIDAR	23
3.4.2. CÁLCULO DE MÉTRICAS LIDAR	23
3.4.3. COMPOSICIÓN DE MOSAICOS LIDAR	25
3.4.4. PREPROCESO DE DATOS DE CAMPO	25
3.4.5. MUESTREO	25
3.4.6. EXTRACCIÓN DE MÉTRICAS	28
3.4.7. ANÁLISIS DE CORRELACIÓN	28
3.4.8. CÁLCULO DE MODELOS DE REGRESIÓN	28
3.5. DESCRIPCIÓN DE HERRAMIENTAS UTILIZADAS	33
3.5.1. QGIS VERSIÓN 3.22.7	33
3.5.2. PYTHON 3.9 Y SPYDER ANACONDA	33
3.5.3. LASTOOLS	33
<b>4. RESULTADOS Y DISCUSIÓN</b>	<b>35</b>
4.1. ANÁLISIS DESCRIPTIVO DE LOS DATOS DE CAMPO	35
4.2. MAPAS	37
4.2.1. ESTADÍSTICA DESCRIPTIVA	39
4.3. CORRELACIONES	49
4.3.1. PEARSON	50
4.3.2. KENDALL	51
4.3.3. SPEARMAN	52
4.4. MODELOS DE REGRESIÓN	53
4.4.1. F. SYLVATICA	54
4.4.2. P. SYLVESTRIS	56
<b>5. CONCLUSIONES</b>	<b>59</b>
<b>6. BIBLIOGRAFÍA</b>	<b>60</b>
<b>7. ANEXOS</b>	<b>64</b>
I. ANEXO I	64
→ Composición de Mosaicos LiDAR	64
II. ANEXO II	65
→ Filtrado de los datos de campo	65
III. ANEXO III	67
→ Extracción métricas LiDAR	67
IV. ANEXO IV	68
→ Cálculo de métricas	68
V. ANEXO V	73
→ Cálculo modelos de regresión	73
VI. ANEXO VI	76
→ Muestreo de F.Sylvatica	76
VII. ANEXO VII	78



---

→	<i>Compuestos anuales de Sentinel-1 y Sentinel-2:</i> .....	78
VIII.	ANEXO VIII:.....	80
→	<i>Descarga y preproceso LiDAR:</i> .....	80
IX.	ANEXO IX: .....	87
→	<i>Mapas de los mosaicos de las métricas LiDAR:</i> .....	87

## ÍNDICE DE FIGURAS

Figura 1: Área de estudio.....	14
Figura 2: Diagrama de flujo del proceso de este trabajo .....	15
Figura 3: Malla de las hojas LiDAR.....	22
Figura 4: Muestra de elevación media en el entorno del monte Beriain-San Donato.....	24
Figura 5: Muestra puntos: los puntos morados corresponden a los píxeles de 20x20 y los puntos amarillos corresponden a los píxeles de 10x10m .....	26
Figura 6 : Muestra de cómo el buffer generado solo cubre un único punto de los cuatro: .....	27
Figura 7: Muestreo realizado para <i>F.sylvatica</i> .....	28
Figura 8: Ejemplo de <code>n_estimators</code> (Saxena, 2020).....	30
Figura 9: Ejemplo de <code>max_features</code> .....	30
Figura 10: Ejemplo de <code>max_depth</code> .....	31
Figura 12: Histogramas de las variables de interés de <i>F.sylvatica</i> respecto a los datos de ordenaciones forestales .....	36
Figura 13: Histogramas de las variables de interés de <i>P.sylvestris</i> respecto a los datos de ordenaciones forestales .....	37
Figura 13: Mosaico de la densidad LiDAR.....	38
Figura 14: Mosaico de la Elevación media (m) LiDAR.....	39
Figura 15: Histogramas de las métricas LiDAR para <i>F.sylvatica</i> .....	44
Figura 16: Histogramas de las métricas LiDAR calculadas para <i>P.sylvestris</i> .....	48
Figura 17: Correlaciones de las métricas LiDAR para <i>F.sylvatica</i> , método Pearson.....	50
Figura 18: Correlaciones de las métricas LiDAR para <i>F.sylvatica</i> , método Kendall.....	51
Figura 19: Correlaciones de las métricas LiDAR para <i>F.sylvatica</i> , método Spearman.....	52
Figura 20: Correlaciones de las métricas LiDAR para <i>P.sylvestris</i> , método Pearson .....	105
Figura 21: Correlaciones de las métricas LiDAR para <i>P.sylvestris</i> , método Kendall.....	106
Figura 22: Correlaciones de las métricas LiDAR para <i>P.sylvestris</i> , método Spearman ...	107

## ÍNDICE DE TABLAS

Tabla 1: Estructura del archivo shape de masas facilitada de las ordenaciones forestales. ....	16
Tabla 2: Estructura del archivo shape de rodales facilitada de las ordenaciones forestales. ....	17
Tabla 3: Estructura de la tabla de aprovechamientos de las ordenaciones forestales. ....	18
Tabla 4: Estructura de la tabla de especies de las ordenaciones forestales. ....	19
Tabla 5: Estructura de la tabla de montes de las ordenaciones forestales. ....	19
Tabla 6: Características del segundo vuelo LiDAR de Navarra, año 2017. ....	21
Tabla 7: Clasificación de puntos LiDAR en la segunda cobertura. ....	22
Tabla 8: Listado de métricas LiDAR calculadas. ....	23
Tabla 9: Bandas de los Satélites Sentinel-1 y Sentinel-2 utilizadas. ....	32
Tabla 10: Herramientas de LAStools utilizadas. ....	34
Tabla 11: Estadística descriptiva de los datos de las ordenaciones forestales. ....	35
Tabla 12: Métrica densidad y métricas LiDAR relacionadas con la elevación para la especie <i>F.sylvatica</i> , parte 1. ....	40
Tabla 13: Métricas LiDAR relacionadas con la elevación para la especie <i>F.sylvatica</i> , parte 2. ....	40
Tabla 14: Métricas LiDAR relacionadas con la intensidad para la especie <i>F.sylvatica</i> . ....	41
Tabla 15: Métricas LiDAR relacionadas con la forma para la especie <i>F.sylvatica</i> . ....	41
Tabla 16: Métricas LiDAR relacionadas con la elevación para la especie <i>P.sylvestris</i> . ....	45
Tabla 17: Métricas LiDAR relacionadas con la intensidad para la especie <i>P.sylvestris</i> . ....	45
Tabla 18: Métricas LiDAR relacionadas con la forma para la especie <i>P.sylvestris</i> , parte 1. ....	46
Tabla 19: Métricas LiDAR relacionadas con la forma para la especie <i>P.sylvestris</i> , parte 2. ....	46
Tabla 20: Resultados de la precisión de los modelos de regresión para el área basimétrica obtenidos con <i>F.sylvatica</i> . ....	54
Tabla 21: Mejores parámetros para el árbol de decisión de área basimétrica de <i>F.sylvatica</i> . ....	54
Tabla 22: Resultados de la precisión de los modelos de regresión para la altura dominante obtenidos con <i>F.sylvatica</i> . ....	55
Tabla 23: Mejores parámetros para el árbol de decisión de altura dominante de <i>F.sylvatica</i> . ....	55
Tabla 24: Resultados de la precisión de los modelos de regresión para volumen total obtenidos con <i>F.sylvatica</i> . ....	56
Tabla 25: Mejores parámetros para el árbol de decisión de volumen de <i>F.sylvatica</i> . ....	56
Tabla 26: Resultados de la precisión de los modelos de regresión para el área basimétrica obtenidos con <i>P. sylvestris</i> . ....	56
Tabla 27: Mejores parámetros para el árbol de decisión de área basimétrica de <i>P.sylvestris</i> . ....	57
Tabla 28: Resultados de la precisión de los modelos de regresión para la altura dominante obtenidos con <i>P. sylvestris</i> . ....	57
Tabla 29: Mejores parámetros para el árbol de decisión de altura dominante de <i>P.sylvestris</i> . ....	57

---

Tabla 30: Resultados de la precisión de los modelos de regresión para el volumen obtenidos con <i>P. sylvestris</i> .....	58
Tabla 31: Mejores parámetros para el árbol de decisión de volumen de <i>P.sylvestris</i> .....	58

## PRÓLOGO

Este trabajo es el trabajo fin de máster del Máster Universitario en Sistemas de Información Geográfica y Teledetección impartido por la Escuela Técnica Superior de Ingeniería Agronómica y Biociencias (ETSIAB) de la Universidad Pública de Navarra (UPNA) y ha sido elaborado durante el curso 2021-2022.

El trabajo ha sido llevado a cabo en parte gracias a la concesión de una beca de colaboración (aprobada por la Resolución N° 1638/2022) con el departamento de Ingeniería de la Universidad Pública de Navarra para el proyecto de "Nuevas herramientas para la gestión forestal basadas en el uso de tecnologías aéreas y satelitales de observación de la tierra (forestOBS)" en la que trabajan en conjunto la Asociación Forestal de Navarra (FORESNA-ZURGAIA) y la UPNA.

## NOTACIÓN

---

AD	Altura Dominante
AB	Área Basimétrica
R2	Coefficiente de determinación $R^2$
LiDAR	Light Detection and Ranging
RMSE	Raíz del error cuadrático medio
SAR	Radar de apertura sintética
RF	Random Forest
VT	Volumen Total

## 1. INTRODUCCIÓN

Históricamente la adquisición de datos sobre la estructura de las masas forestales ha estado ligada al trabajo de campo. Esto ha conllevado a necesitar abundante tiempo para su captura y muchas veces quedarse limitado en no poder llegar a cubrir toda un área, cuando se trata, por ejemplo, de zonas extensas y/o remotas (Bergen & Dronova, 2007).

Con el avance de la cantidad y diversidad de imágenes aéreas libres y abiertas (Wulder & Coops, 2014) se ha visto que la teledetección puede ser una herramienta de gran utilidad para la gestión forestal. Teniendo en cuenta la revolución en la recolecta y manipulación de sus datos en los últimos años (Gorelick et al., 2017), la teledetección ha abierto un abanico de posibilidades dentro de la silvicultura, optimizando, por ejemplo, el trabajo de campo y aportando parámetros estructurales necesarios de las masas (Roberts et al., 2007). Sus productos derivados dan la oportunidad de evaluar el estado de las masas forestales y así, poder generar documentos oficiales que establezcan la base de su correcta gestión y conservación, como lo son los inventarios forestales (Puliti et al., 2021).

Dentro del ámbito de la teledetección, son múltiples los sensores que han sido utilizados para la estimación de datos forestales, tanto sensores pasivos como activos; sensores ópticos, multiespectrales, radar o LiDAR. La gran diferencia entre estos sensores se basa en que los sensores pasivos miden la energía o radiación solar tras su reflexión en las superficies (en este caso forestales) de interés, a diferencia de los activos, que miden el pulso de energía emitido por ellos mismos, después de haber sido reflejado en la superficie (Chuvienco, 1995). A pesar de que los sensores pasivos quedan limitados por la disponibilidad de radiación solar y por efectos como la presencia de nubes o la dispersión atmosférica (Shugart et al., 2010) no quiere decir que no sean tan efectivos para la gestión forestal, ya que sensores como SPOT, Landsat, Sentinel-2, IKONOS y/o GeoEye-1, han demostrado ser efectivos para desarrollar inventarios forestales en todo el mundo (Castillo-Santiago et al., 2010) (Puliti et al., 2021).

Por otra parte, sensores activos como los sensores radar (SAR) y entre ellos Sentinel-1, han dado buenos resultados a la hora de medir la altura del dosel forestal (Huang et al., 2022). Este sensor puede estimar la estructura del bosque con alta resolución espacial (25–100 m) a partir de la retrodispersión observada en la región de las microondas (Hall et al., 2011). Sin embargo, un problema común a sensores ópticos y SAR es su eventual saturación a valores altos de biomasa. En concreto, los sensores SAR quedan limitados cuando los niveles de biomasa oscilan entre 10 y 150mg/ha tanto en la banda L como en la banda P (con valores algo superiores), ya que pierden sensibilidad por saturación (Houghton et al., 2009) (Shugart et al., 2010).

Otro de los sensores remotos más utilizados para gestión forestal y que a día de hoy son los que están atrayendo más interés en este ámbito, son los sensores LiDAR (Hudak et al., 2009). Estos sensores son una tecnología activa, que mediante emisiones múltiples de haces de luz (o pulsos) y el registro de sus posteriores retornos, pueden estimar directamente la altura y la estructura vertical de los bosques (Lefsky et al., 2002) gracias a que son capaces de proporcionar medidas de alta precisión sobre la geometría de los objetos observados (coordenadas x, y, z) y la intensidad de los pulsos de retorno (Yu et al., 2010). Los datos LiDAR tienen grandes ventajas para estimar variables de interés

forestal. Por ejemplo, pueden capturar la estructura tridimensional del bosque (Lim et al., 2003) y dar lugar a modelos que estimen la biomasa incluso en bosques tropicales donde las estructuras son densas y complejas (Coomes et al., 2017) (Phua et al., 2016). Otros estudios concluyen haber estimado el volumen forestal a gran escala y con una eficiencia temporal óptima (Hu et al., 2021). También han demostrado ser capaces de estimar los estados de desarrollo de los árboles (Falkowski et al., 2009), así como los procesos de crecimiento de hojas nuevas y la pérdida por abscisión (caída de hojas y ramas) (Calders et al., 2015).

Con todo ello, el LiDAR se ha convertido en uno de los sensores más útiles para caracterizar la estructura forestal (Zimble et al., 2003) y en una poderosa herramienta en la creación de inventarios forestales modernos (Wallace et al., 2012). Por ello, son muchos los que se han sumado a la investigación y desarrollo de metodologías de estimación de variables forestales a través de la tecnología LiDAR, que han dado lugar a una profundización en esta materia y la publicación de numerosos estudios. Por ejemplo, en Noruega se han realizado estudios en los cuales se utilizó LiDAR para la extracción de variables forestales que daban conocimiento de la altura y densidad de las masas forestales, basados en el desarrollo de modelos de regresión lineal múltiple con transformación logarítmica (Næsset & Økland, 2002). En los bosques tropicales de la Estación Biológica de la Selva (Costa Rica), se utilizaron datos LiDAR para medir la estructura forestal y la dinámica de la biomasa mediante regresión lineal múltiple (Dubayah et al., 2010). En Sao Paulo (Brasil), con un modelo lineal se estimó la biomasa de los troncos en las plantaciones de eucaliptos (Silva et al., 2015).

Por tanto, parece que el LiDAR es una de las opciones más favorables para estimar variables forestales, aunque conviene recordar que otros tipos de sensores pueden también ser útiles y eficientes en ello. Una estrategia interesante a explorar es la de fusionar datos adquiridos por múltiples sensores, y evaluar y analizar los eventuales beneficios de la combinación de estos, ya que algunos autores aseguran poder mejorar los niveles de precisión alcanzados con cada fuente de datos de forma individual (Zolkos et al., 2013).

Los artículos antes citados, utilizan en muchos casos modelos lineales para hacer estimaciones de las variables de interés forestal, aunque algunas investigaciones (Ashraf et al., 2013) sugieren que sería mejor utilizar otro tipo de modelos. Ya que, las variables biofísicas que afectan al crecimiento de los árboles, como la luz, la temperatura, el agua del suelo o las condiciones de los nutrientes son difíciles de incorporar a esos modelos por sus complejas relaciones no lineales. Por ello existen autores que utilizan modelos no lineales en este tipo de trabajos, entre ellos, uno de los que más destaca es el algoritmo Random Forests (RF) (Breiman, 2001), que ha sido utilizado en diversas investigaciones enfocadas a estimar variables forestales. En un estudio realizado en Finlandia se estimaron la altura media, el diámetro medio y el volumen a partir de varios algoritmos, entre los cuales RF fue el que menor error (RMSE) dio (Yu et al., 2010).

No son pocos los autores que destacan la eficacia de RF para estimar variables forestales y no solo en este tipo de estimaciones cuantitativas, sino también, para realizar clasificaciones de especies de árboles. En el siguiente estudio, el objetivo del trabajo era conseguir clasificar el estrato del bosque y RF reveló la mayor precisión de clasificación entre los todos enfoques de clasificación probados (Lineal Discriminant, Naïve Bayes,

Support Vector Machines, K-vecino más cercano, Neural Networks y Random Forests), con una precisión general media del 83,3 % (Wernicke et al., 2022). En el siguiente estudio realizado en la región forestal de Puer, provincia de Yunnan (China), por una parte, los datos LiDAR ayudaron a mejorar la precisión de las clasificaciones de especies (Pang et al., 2016) y por otra, RF fue quien clasificó con mejor precisión. Por último, en otro estudio también realizado en China, en la provincia de Hunan, con el objetivo de mapear la altura de los bosques mediante datos satelitales de múltiples fuentes, el mejor resultado se obtuvo con RF frente a otro tipo de modelos como la regresión multilínea y regresión de vector de soporte. (Huang et al., 2022).

Esta breve revisión demuestra, en primer lugar, que la tecnología LIDAR tiene un interés potencial de cada a la estimación de variables forestales, tales como la biomasa, la estructura de copas, el área, el volumen o la altura. En segundo lugar, se constata que uno de los algoritmos más eficaces para realizar modelos de regresión en este ámbito es RF.

En consecuencia, el presente trabajo trata de evaluar la capacidad de los datos LiDAR adquiridos sobre Navarra en 2017 para estimar variables forestales en masas típicas de esta región. Las variables forestales estudiadas son tres: el área basimétrica, la altura dominante y el volumen total, y se consideran dos tipos de masas: el haya (frondosa) y el pino silvestre (conífera). Con este fin, se calculan distintas métricas del vuelo LiDAR y se ajustan distintos modelos RF con las variables de interés. También se comparan los resultados obtenidos a partir de modelos basados en datos LiDAR con otras fuentes de datos, en concreto las imágenes adquiridas por los satélites Sentinel-1 y Sentinel-2, y se evalúa también el uso de modelos combinados, para poder concluir qué fuente de información es la más precisa para estimar las variables de interés.

Finalmente, se quiere subrayar que a pesar de que este tipo de análisis pueden entenderse como orientados a suplantar el trabajo de campo, es necesario recordar que la obtención de este tipo de modelos, en todos los casos requiere de datos de campo de entrenamiento, con lo que en lugar de suplantarlo este tipo de herramientas pueden ser de utilidad para aprovechar mejor el trabajo de campo realizado y para entender mejor la realidad de las masas forestales en terrenos amplios, remotos y difícilmente abarcables. También se quiere destacar, que este trabajo constituye una primera evaluación y su objeto no es aún proponer ninguna metodología operativa, para lo que se necesita profundizar y ampliar el análisis. De todas maneras, se espera que el presente estudio contribuya al desarrollo futuro de herramientas operativas.



## 2. OBJETIVOS

El objetivo general de este trabajo fin de máster es generar modelos de regresión que estimen las variables forestales área basimétrica, altura dominante y volumen total de dos especies forestales, *F. sylvatica* y *P. sylvestris*, de Navarra mediante métricas obtenidas a partir de la tecnología LiDAR.

### 2.1. OBJETIVOS ESPECÍFICOS

Para alcanzar este objetivo general se plantean los siguientes objetivos específicos:

- Seleccionar métricas LiDAR potencialmente útiles para la estimación de las variables forestales a estudio
- Calcular las métricas LiDAR seleccionadas para todas las hojas del vuelo LiDAR realizado sobre Navarra en 2017 y realizar mosaicos para toda Navarra.
- Realizar un análisis estadístico descriptivo de las métricas LiDAR calculadas incluyendo un análisis de la correlación entre métricas.
- Seleccionar las métricas óptimas para la estimación de las variables forestales de interés: área basimétrica, altura dominante y volumen total.
- Crear modelos de regresión RF que permitan estimar las variables forestales de interés.
- Comparar los modelos de regresión obtenidos a partir de LiDAR con otros obtenidos a partir de imágenes satelitales Sentinel-1 y Sentinel-2 y evaluar los beneficios de la combinación de distintos tipos de datos.

### 3. MATERIAL Y MÉTODOS

#### 3.1. ÁREA DE ESTUDIO

El área de estudio la representan las masas forestales de la Comunidad Foral de Navarra, en concreto, una serie de polígonos de estudio correspondientes a masas ordenadas fueron facilitados por el Departamento de Desarrollo Rural y Medio Ambiente del Gobierno de Navarra (Figura 1: Área de estudio): 11.124 masas (polígonos) que cubren 784,073km<sup>2</sup>.

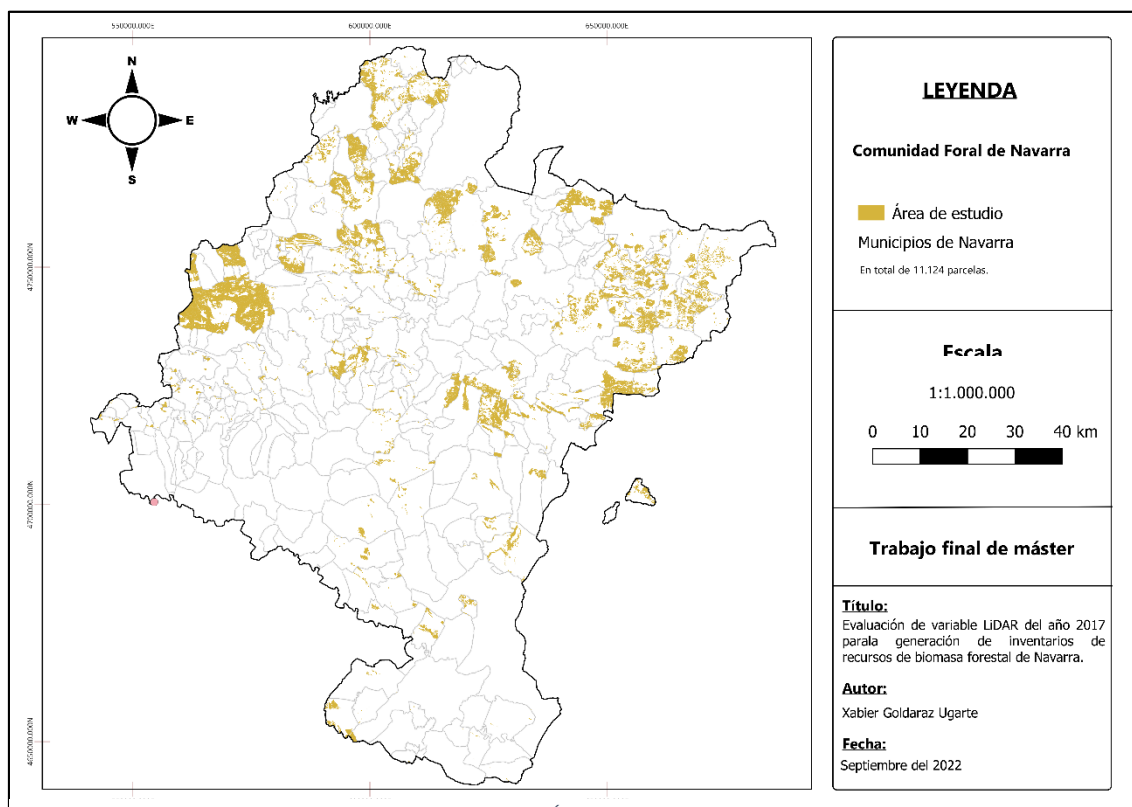


Figura 1: Área de estudio

#### 3.2. FLUJO DE TRABAJO

Para exponer de forma clara el apartado de material y métodos se aporta un diagrama de flujo de trabajo en el que se reflejan tanto los materiales utilizados (datos y herramientas) como los pasos seguidos en la metodología para conseguir los resultados definidos en los objetivos.

El desarrollo de este Trabajo Fin de Máster se puede dividir en tres fases genéricas: inicio, proceso y resultados (Figura 2).

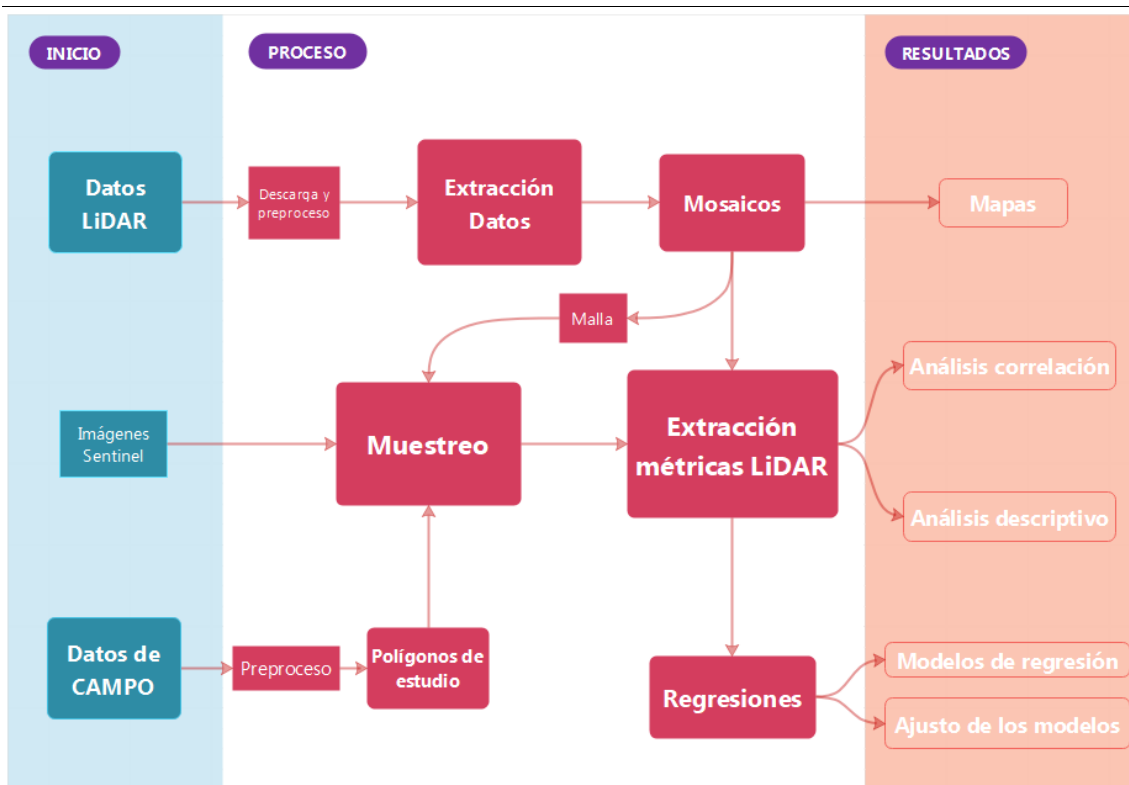


Figura 2: Diagrama de flujo del proceso de este trabajo

El trabajo cuenta con dos tipos de datos que son la base de este trabajo: los datos LiDAR adquiridos sobre Navarra en 2017 y los datos de campo de los proyectos de ordenación forestal referidos en el punto anterior. A estos datos se añade un tercer conjunto de datos constituido por las imágenes satelitales Sentinel-1 y Sentinel-2 adquiridas sobre Navarra en 2018.

Los datos LiDAR, constituyen un conjunto de datos publicado en abierto en el Repositorio de Descarga de Cartografía del Gobierno de Navarra (<https://filescartografia.navarra.es/>), posteriormente en el apartado de metodología (3.4) se desarrollan sus detalles. En cuanto a los datos de campo, tal y como se ha expuesto en el punto anterior, fueron facilitados por el Departamento de Desarrollo Rural y Medio Ambiente del Gobierno de Navarra para el desarrollo de esta investigación, aunque por el momento no representan un conjunto de datos publicado en abierto. Las imágenes satelitales Sentinel-1 y Sentinel-2 forman parte del programa europeo de observación de la tierra Copernicus, cuya distribución es también libre y gratuita.

Si bien en los siguientes apartados se explican en detalle los diferentes materiales y los métodos aplicados, en las siguientes líneas se aporta una breve descripción del diagrama de flujo aportado en la (Figura 2). El proceso comienza con los datos LiDAR, que tras su descarga son sometidos a un preproceso y un cálculo de métricas de potencial interés forestal que concluye con la creación de mosaicos a escala de Navarra que *per se* constituyen un resultado de gran interés.

Por otra parte, los datos de campo son también sometidos un preproceso para obtener, filtrar y seleccionar los polígonos de trabajo.

Con estos dos conjuntos de datos y la colección de imágenes Sentinel-1 y Sentinel-2 del año 2018 (procesada por el grupo de investigación THERRAE de la UPNA) se realiza un muestreo que constituirá la base para el análisis descriptivo de las métricas LiDAR obtenidas y la creación de modelos de regresión para estimar variables forestales (altura dominante, área basimétrica y volumen total) a partir de métricas LiDAR, datos Sentinel-1, y/o Sentinel-2.

Tanto para la fase de análisis descriptivo como para la de modelado se realiza una extracción de estadísticas de las colecciones de datos LiDAR, Sentinel-1 y Sentinel-2 para los puntos seleccionados en el muestreo. Esta extracción de datos es la base para el análisis descriptivo de los datos, el análisis de correlación y para entrenar y validar los modelos de regresión ajustados.

### 3.3. MATERIALES (Inicio):

#### 3.3.1. DESCRIPCIÓN DE DATOS DE ORDENACIONES FORESTALES

Los datos de campo de ordenaciones forestales facilitados por el Gobierno de Navarra se estructuran en dos capas en formato shape (SHP) y tres archivos en formato Excel.

Los archivos SHP llamados "f\_masas" y "f\_rodales" comprenden:

- Masas: Constituye el archivo principal del que se extraen los datos de campo en que se basa este trabajo. Comprenden datos relacionados con la tipología forestal de cada polígono (masa forestal ordenada).

Cada masa cuenta con un total de 21 atributos (datos en columnas). Tiene un identificador específico de la masa (id) un identificador secundario (IDPOM), su uso, la especie principal (SP1), secundaria y terciaria (SP2, SP3), la fracción que corresponde a cada especie (FCCSP1, FCCSP2, FCCSP3), características de la masa (calidad de estación, año, daños, descripción), variables forestales (área basimétrica, altura dominante, volumen total y madera muestreable) y la geometría.

En la siguiente Tabla 1 podemos observar su estructura.

*Tabla 1: Estructura del archivo shape de masas facilitada de las ordenaciones forestales.*

ATRIBUTO	DESCRIPCIÓN
<b>id</b>	
<b>IDPOM</b>	Identificador del Instrumento de Planificación Forestal
<b>Masa</b>	
<b>Uso</b>	1 Arbolado 2 Matorral 3 Pasto 4 Inforestal
<b>Tipo</b>	1 Adultas susceptibles de corta final 2 Adultas productos leñosos 3 Adultas claras 4 Jóvenes clareos 5 Monte medio o bajo leñoso 6 Heterogéneas 7 Irregulares 8 Bosques de ribera 9 No arbolada 10 Masas de arbolado disperso y uso silvopastoral
<b>SP1</b>	Código de especie principal

<b>FCCSP1</b>	Fracción Cabida Cubierta especie 1. (%)
<b>SP2</b>	Código de especie secundaria
<b>FCCSP2</b>	Fracción Cabida Cubierta especie 2. (%)
<b>SP3</b>	Código de especie terciaria
<b>FCCSP3</b>	Fracción Cabida Cubierta especie 3. (%)
<b>FCC</b>	Fracción Cabida Cubierta total. (%)
<b>Calidad de Estación</b>	Buena / Regular / Mala
<b>Edad</b>	Edad en el año de inventario o intervalo de edades para masas no procedentes de repoblación. Se trata de clases de edad artificial.
<b>Área Basimétrica</b>	m <sup>2</sup> /Ha
<b>Altura Dominante</b>	Metros
<b>Volumen Total</b>	Volumen total con corteza en m <sup>3</sup> /Ha
<b>Madera Muerta</b>	Código, cualitativo frente a la abundancia normal encontrada en el Inventario Forestal Nacional para cada especie
<b>Daños</b>	2 opciones: 0=No, 1=Sí (son reseñables).
<b>Descripción</b>	Para indicar detalles relevantes como pedregosidad, mezcla de especies etc.
<b>geometry</b>	Coordenadas X e Y.

Existen masas en las que no hay ningún dato de las variables forestales de interés (Área basimétrica, Altura dominante y Volumen total), es por eso, que se realiza un filtrado para excluir del análisis los recintos que no cuentan con esa información (ver apartado 3.4.5).

- **Rodales:** Es la unidad de gestión forestal básica. Una masa está compuesta por varios rodales. El rodal se puede definir como un área variable, pero con constancia de las características de masa y estación, y por tanto de tratamiento (Serrada,1995). Los atributos de este archivo SHP se muestran en la Tabla 2.

*Tabla 2: Estructura del archivo shape de rodales facilitada de las ordenaciones forestales.*

<b>ATRIBUTO</b>	<b>DESCRIPCIÓN</b>
<b>Id</b>	Identificador
<b>Idpom</b>	Identificador del Instrumento de Planificación Forestal
<b>Cuartel</b>	Letra o número de cuartel (vacío si no existen)
<b>Cantón</b>	Número o nombre
<b>Rodal</b>	Letra del rodal o recinto en minúsculas. Todos los rodales deben llevar letra, incluso los rodales únicos por cantón.
<b>Masa</b>	Nº de masa.
<b>Idrodal</b>	Identificador del rodal
<b>Titular</b>	Titularidad del recinto.
<b>Área</b>	Área del rodal.
<b>Geometry</b>	Geometría del polígono

Este archivo de rodales no va a ser tenido en cuenta en este trabajo, puesto que no aporta ninguna información adicional de interés para los objetivos planteados.

Por otra parte, se cuenta con los archivos Excel "tabla de monte", "tabla de aprovechamientos" y "tabla de especie":

- Tabla de aprovechamientos: Hace referencia a los rodales y se detallan los aprovechamientos autorizados, el tipo de ejecuciones, fecha, especie, superficie. La estructura de esta tabla tiene los atributos que se muestran en la Tabla 3

*Tabla 3: Estructura de la tabla de aprovechamientos de las ordenaciones forestales.*

ATRIBUTO	DESCRIPCIÓN
<b>id</b>	Identificador
<b>idpom</b>	Identificador del Instrumento de Planificación Forestal
<b>Rodalid</b>	Identificador del rodal.
<b>Tipo de actuación</b>	1 Corta a hecho 2 Aclareo sucesivo 3 Entresaca 4 Corta por huroneo 5 Corta de mejora 6 Corta mixta
<b>Subtipo</b>	1 Corta Final 2 Corta Aclaratoria 3 Corta Diseminatoria 4 Corta Preparatoria 5 Clara selectiva 6 Clara sistemática 7 Corta sanitaria 8 Corta silvopastoral 9 Otra clara
<b>Continuidad</b>	1 Uniforme 2 Bosquetes 3 Fajas
<b>Superficie de actuación</b>	Superficie en ha
<b>Especie</b>	Código de especie
<b>Características</b>	Descripción de la actuación programada Es el lugar para explicar si la masa es mixta y el valor de especie es relativo.
<b>Año previsto de actuación</b>	Fecha
<b>Volumen total a extraer</b>	Volumen total de madera en m3. El valor se da por rodal
<b>Volumen maderable a extraer</b>	Volumen maderable en m3 por rodal
<b>Calidad de fustes</b>	
<b>Año de ejecución</b>	Fecha
<b>Ejecutado</b>	Verdadero/Falso

núm. expte	
Observaciones	Para describir el seguimiento

- Tabla de especie: En esta tabla se detalla el código con el que se identifica cada especie, su nombre científico y su nombre común (Tabla 4). Comprende un total de 171 especies.

*Tabla 4: Estructura de la tabla de especies de las ordenaciones forestales.*

ATRIBUTO	DESCRIPCIÓN
<b>Id</b>	Identificador
<b>Nombre común</b>	Nombre común de la especie
<b>Código de especie</b>	Código (número) de la especie
<b>Especie</b>	Nombre científico

- Tabla de monte: En este archivo se aportan datos sobre los proyectos de ordenación forestal realizados, indicando su año de ejecución, la resolución, el gabinete forestal que ha realizado el proyecto, etc. (Tabla 5)

*Tabla 5: Estructura de la tabla de montes de las ordenaciones forestales*

ATRIBUTO	DESCRIPCIÓN
<b>idpom</b>	Identificador del Instrumento de Planificación Forestal
<b>Nombre del Plan</b>	Nombre del Instrumento de Planificación Forestal
<b>Monte</b>	Nombre del monte
<b>Año de redacción</b>	Fecha
<b>Gabinete Forestal</b>	Gabinete Forestal redactor
<b>Resolución de aprobación</b>	Resolución
<b>Titularidad</b>	0 Comunal 1 Privado 2 P.F.N.
<b>Pertenencia</b>	
<b>Comarca forestal</b>	Pirenaica / Atlántica / Mediterránea
<b>Año primera ordenación</b>	Fecha
<b>Número de revisión</b>	
<b>MUP</b>	Número de montes catalogado de Utilidad Pública
<b>Vigencia</b>	

Entre todos estos datos disponibles, para la realización de este trabajo únicamente se utilizará el archivo de Masas y los códigos de especie. El resto de los archivos se utilizan como información adicional pero no forman el núcleo de los datos utilizados. El archivo de Masas contiene los datos de las variables de estudio; Área basimétrica, Altura dominante y Volumen total, que se exponen a continuación

→ *Área basimétrica:*

El área basimétrica, también denominada Área Basal, y que, en terminología dasonómica, se representa por  $G$ , es una de las variables de referencia de la masa forestal, cuyos valores son universalmente utilizados en la gestión de su espesura. Este parámetro resulta de expresar en  $m^2/ha$ , la relación entre las secciones normales de los árboles de un espacio forestal y la superficie de terreno que ocupan.

Para su cálculo debemos medir en una o varias superficies muestrales, los diámetros normales de todos los pies mayores presentes en ellas, lo que nos permitirá obtener la superficie de sus secciones supuestas circulares, y referir la suma de todas ellas en  $m^2/ha$  (McElhinny et al., 2005).

La sección normal de un árbol es la que se encuentra a la altura del pecho, o a 1,30 m de su base. El valor de su superficie supuesta circular, que se suele expresar como "g", se obtiene a través de la medida de su diámetro "d" (McElhinny et al., 2005):

→ *Altura dominante:*

Este parámetro se define como la altura media de los 100 pies más gruesos por hectárea (Assmann, 1970). Queda expresada en metros (m).

→ *Volumen total:*

El volumen total se corresponde con el volumen de madera existente en  $m^3$  por hectárea.

### 3.3.2. DESCRIPCIÓN DE DATOS LIDAR

En este trabajo se han utilizado los datos de la segunda cobertura LiDAR adquirida sobre Navarra en el año 2017. Este vuelo LiDAR se realizó en el marco del Plan Nacional de Ortofotografía Aérea, uno de cuyos objetivos es cubrir todo el territorio de España cada seis años mediante nubes de puntos con coordenadas X, Y, Z y atributos como clasificación o color, obtenidas mediante sensores LiDAR aerotransportados. En la primera cobertura (entre 2009 y 2015) la densidad de puntos fue de 0,5-1,0 puntos/ $m^2$  y en la segunda entre 0,5-4,0 puntos/ $m^2$  con la excepción de Navarra, que gracias al uso del innovador sensor LEICA SPL100 adquirió datos con una densidad de 14 puntos/ $m^2$ .

La descarga de dichos datos se realizó desde el Repositorio de Descarga de Cartografía del Gobierno de Navarra (<https://filescartografia.navarra.es/>), siendo las características generales de los datos de esta segunda cobertura LIDAR las expuestas en la Tabla 6.



Tabla 6: Características del segundo vuelo LiDAR de Navarra, año 2017.

CARACTERÍSTICAS DEL SEGUNDO VUELO LIDAR	
<b>Sensor</b>	LEICA SPL100
<b>Cámara</b>	RDC 30
<b>Fecha de inicio de vuelo</b>	Septiembre 2017
<b>Fecha de fin de vuelo</b>	Noviembre 2018
<b>Plataforma</b>	Avión Beechcraft B200 King Air
<b>Patrón escaneo</b>	Circular
<b>Laser</b>	100 canales: todos en 532
<b>Retorno múltiple</b>	Hasta 10 por canal
<b>Líneas de vuelo</b>	482
<b>Altura de vuelo</b>	3.900 – 6.300 metros
<b>Ancho de pasada</b>	2.300 metros
<b>Velocidad</b>	200 nudos
<b>Densidad de puntos</b>	14pts/m <sup>2</sup>
<b>Bits</b>	16

Esta segunda cobertura se publica en hojas de 1kmx1km en formato comprimido LAZ siguiendo el formato 1.4 del estándar (ASPRS, 2019). Cada hoja se nombra con una cadena de números y letras que tienen un significado específico:

- Lo 7 u 8 primeros dígitos: detallan si el vuelo ha sido clasificado manualmente o mediante clasificación automática mejorada con procesos de *machine learning* y depuración de resultado.
- Del dígito 8 al 14: Corresponden a las coordenadas X e Y de la esquina superior izquierda en km.
- A continuación, se indica el año seguido de una abreviatura de Navarra (NAV)
- Por último, se indica el Sistema Geodésico de Referencia: EPSG 25830.

Tal y como se ha comentado, los puntos de esta segunda cobertura LiDAR llevan asignada una clasificación, cuya codificación (Tabla 7) es diferente a los de la primera cobertura, dado que esta segunda cobertura se publicó según el formato 1.4 del estándar LAS y la primera según el 1.2.

Tabla 7: Clasificación de puntos LiDAR en la segunda cobertura

NÚMERO	CLASE
2	Suelo
3	Baja vegetación
4	Media vegetación
5	Alta vegetación
6	Edificios
8	Coches
9	Fondo de agua
10	Solape desechado
11	Ruido
12	MDT 2014
17	Puntos fugados
18	Ruido sensor
28	Ruido sensor solape
32	Puentes

El formato LAZ mencionado es un formato estándar de intercambio de nubes de puntos tridimensionales LiDAR creado por la American Society for Photogrammetry and Remote Sensing (ASPRS). Este formato almacena para cada punto de la nube los valores x, y, z, y otro tipo de datos como la intensidad y número de retorno (eco), además de los canales RGB y el tiempo GPS.

Tal y como se ha mencionado, se descargaron y procesaron la totalidad de hojas de 1x1km que cubren la superficie de Navarra. En total 13.161 hojas (Figura 3), que suman un volumen de datos de 5,386 PB de datos.

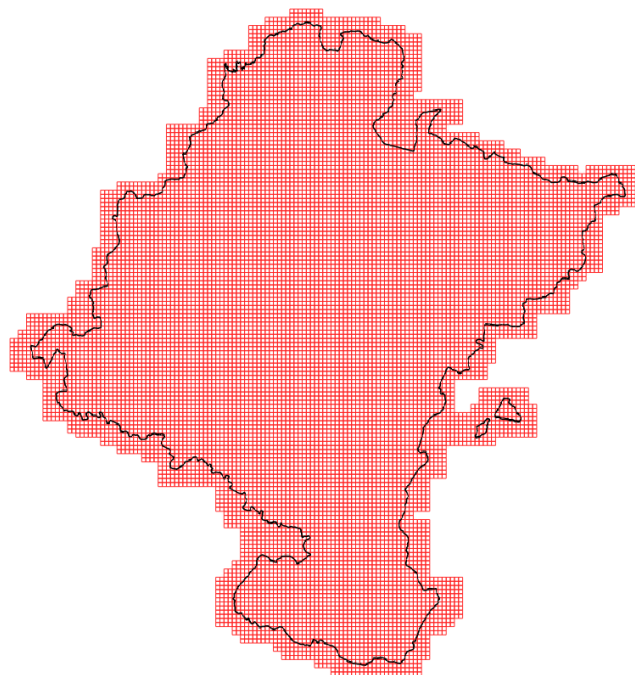


Figura 3: Malla de las hojas LiDAR

### 3.4. METODOLOGÍA (Proceso):

Siguiendo el diagrama de flujo de trabajo (Figura 2), una vez descritos los datos iniciales se pasa a explicar en detalle los principales procesos llevados a cabo en el trabajo.

#### 3.4.1. DESCARGA Y PRETRATAMIENTO DE DATOS LIDAR

El inicio de este trabajo comenzó con el cálculo de métricas LiDAR; es decir, el apartado 3.4.2., a pesar de ello, se ha creído conveniente exponer este apartado para que la metodología tuviese un hilo conductor desde un principio.

El código utilizado para la descarga y pretratamiento de los datos LiDAR se encuentra en el ANEXO VIII.; que ha sido cedido desde el grupo de investigación THERRAE, Departamento de Ingeniería de la Universidad Pública de Navarra.

#### 3.4.2. CÁLCULO DE MÉTRICAS LIDAR

En la Tabla 8 se describen las métricas LiDAR calculadas a partir de las hojas LAZ preprocesadas. Estas métricas fueron seleccionadas dado que en la bibliografía han mostrado una relación potencial con la estructura de la vegetación (Evans et al., 2009).

*Tabla 8: Listado de métricas LiDAR calculadas*

<b>Elevation_average</b>	Elevación media
<b>Elevation_STD</b>	Desviación estándar de la elevación
<b>Elevation_CV</b>	Coefficiente de variación de la elevación
<b>Skewness</b>	Grado de asimetría de la distribución de probabilidad de la elevación
<b>Curtosis</b>	Grado de apuntamiento de la distribución de frecuencias de la elevación
<b>Elevation_lowest</b>	Elevación mínima
<b>Elevation_p25</b>	Percentil 25% de la elevación
<b>Elevation_p50</b>	Percentil 50% de la elevación
<b>Elevation_p75</b>	Percentil 75% de la elevación
<b>Elevation_p90</b>	Percentil 90% de la elevación
<b>Elevation_p95</b>	Percentil 95% de la elevación
<b>Elevation_highest</b>	Elevación máxima
<b>Intensity_average</b>	Intensidad media
<b>Intensity_CV</b>	Coefficiente de variación de la intensidad
<b>Intensity_STD</b>	Desviación estándar de la intensidad
<b>Intensity_lowest</b>	Intensidad mínima
<b>Intensity_highest</b>	Intensidad máxima
<b>Fcc_3m</b>	Fracción de cabida cubierta. Número de puntos de primer retorno por encima de 3 m dividido por el número de puntos de primer retorno totales.
<b>Density</b>	Densidad de puntos por m <sup>2</sup>
<b>Npoints</b>	Número de puntos
<b>Range_c00</b>	Número de puntos cuya altura sobre el suelo esté entre 0 y 3.
<b>Range_c01</b>	Número de puntos cuya altura sobre el suelo esté entre 3 y 10.
<b>Range_c02</b>	Número de puntos cuya altura sobre el suelo esté entre 10 y 15.
<b>Range_c03</b>	Número de puntos cuya altura sobre el suelo esté entre 15 y 20.
<b>Range_c04</b>	Número de puntos cuya altura sobre el suelo esté entre 20 y 25.
<b>Range_c05</b>	Número de puntos cuya altura sobre el suelo esté entre 25 y 30.

**Range\_c06**

Número de puntos cuya altura sobre el suelo esté entre 30 y 35.

\*Todas las elevaciones se calculan en unidades de metros sobre el nivel del suelo.

El código utilizado para calcular las métricas se incluye en el ANEXO IV:

En el primer apartado de este anexo se hace referencia a los directorios en que se estructuran los datos de entrada al proceso. En el segundo apartado se crean las funciones llamando a las herramientas de LAStools que se describen en el apartado 3.5.3. Finalmente, en el tercer apartado y último, se iteran todas las nubes de puntos que se han descargado y preprocesado calculando las 28 métricas de la Tabla 8 para todas ellas, menos las métricas de *elevation\_CV* e *intensity\_CV*, que han sido calculadas mediante la calculadora ráster de QGIS. La fórmula utilizada en la calculadora ha sido la siguiente:

$$Elevation\_CV = Elevation\_STD \div Elevation\_average$$

$$Intensity\_CV = Intensity\_STD \div Intensity\_average$$

El resultado ha sido el cálculo de un archivo ráster (10x10m de píxel) para cada una de 28 métricas por cada hoja LiDAR. El formato elegido ha sido tif (unos 30kB cada uno), generándose también archivos tfw y kml de metadatos, con un peso mucho menor.

En la (Figura 4) se muestra el ráster de elevación media de una hoja de ejemplo.

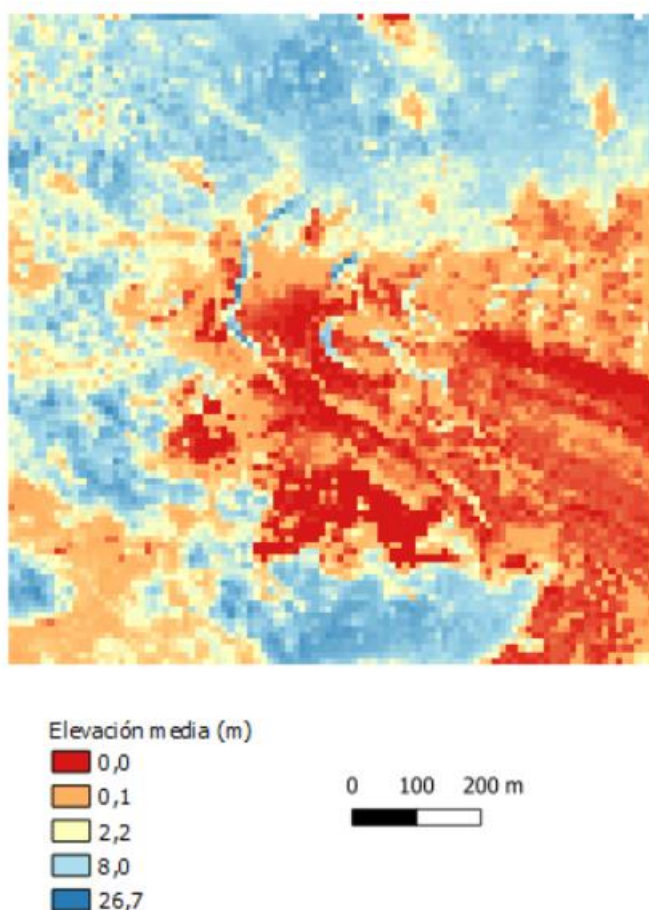


Figura 4: Muestra de elevación media en el entorno del monte Beriain-San Donato.

Realizado este proceso (muy intensivo en cálculo), se ha realizado una revisión de que todas las hojas se hubiesen calculado de manera correcta, que no hubiese ningún error y, lo más importante, que no faltase ninguna de las hojas.

### 3.4.3. COMPOSICIÓN DE MOSAICOS LIDAR

Una vez calculadas las métricas, se realizan mosaicos con los rásteres de todas las hojas de 1x1 km creadas para cada métrica, de forma que se genera un mosaico de la misma para la totalidad de Navarra. El código utilizado para el mosaico de todas las hojas se incluye en el ANEXO I. Este código, parte de una lista de las métricas calculadas, para luego recorrer todas las hojas de cada métrica por su nombre y poder así poder unirlas.

El resultado de este proceso ha sido la creación de 27 mosaicos que cubren la totalidad del territorio de la Comunidad Foral de Navarra, cada uno de ellos con un peso cercano a 1GB.

#### 3.4.3.1. *Estadística descriptiva*

Una vez creados los mosaicos para cada métrica, se calculan una serie de estadísticos descriptivos básicos a modo exploratorio, además de su histograma:

- Valor mínimo
- Valor máximo
- Media
- Mediana
- Desviación estándar

### 3.4.4. PREPROCESO DE DATOS DE CAMPO

En este preproceso de los datos de campo, lo que se pretende es eliminar los polígonos que no cuentan con datos de área basimétrica, altura dominante o volumen total, es decir, desechar las masas que contengan "NaNs". Por otra parte, se quiere obtener las especies principales de las masas y su área (km<sup>2</sup>) y finalmente calcular los datos totales que existen por especie y sus estadísticos: media, desviación estándar, mínimo, P25%, P50%, P75% y máximo.

El código utilizado en este apartado (ANEXO II:) se divide en 7 partes. La primera realiza un primer filtrado y elimina los recintos sin datos. La segunda calcula el área en km<sup>2</sup> de cada polígono. En la siguiente, se reemplaza el código numérico de cada especie por su nombre común. A continuación, se calcula el área total de cada especie. Finalmente se suma esta columna a los datos de estadísticos generales obtenidos para cada especie y se exporta como .xlsx

### 3.4.5. MUESTREO

Partiendo de todos los recintos preprocesados en el apartado anterior, es necesario realizar un muestreo de los mismos para poder obtener una serie de puntos con los que se construirá un archivo de entrenamiento y otro de validación (con métricas LiDAR, datos Sentinel-1 y Sentinel-2 y datos de campo de altura dominante, área basimétrica y volumen total). Estos archivos constituirán la base para el desarrollo de los modelos de regresión y el cálculo de sus métricas de rendimiento o ajuste.

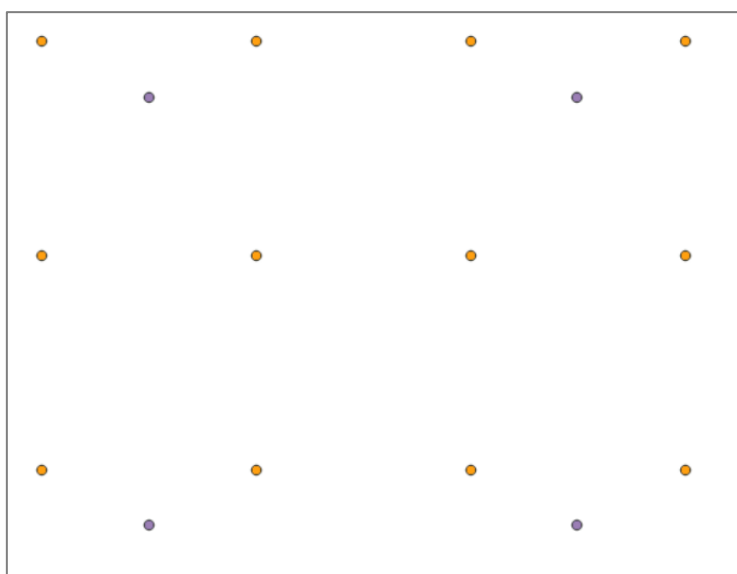
Partiendo de los polígonos preprocesados, se genera una malla de puntos que cubra todos los polígonos. Primeramente, mediante la herramienta de QGIS "Conversion:

Rasterizar" se rasterizarán los polígonos a un tamaño de píxel 10x10 m (la misma que los mosaicos LiDAR). A continuación, mediante la herramienta de QGIS "Píxeles ráster a puntos" se transforma el ráster a archivo de puntos, generando un punto por cada píxel. Con ello se obtiene una malla con un total de 7.831.311 puntos.

Este proceso se repite para el tamaño de píxel de 20x20 m, el de las imágenes Sentinel-1 y Sentinel-2 utilizadas, obteniendo una malla con 1.956.114 puntos. Para realizar modelos de regresión que combinen las métricas LiDAR con los datos de las imágenes es necesario contemplar en el muestreo su diferente resolución y su superposición, es decir, es necesario que haya un único punto de muestreo por cada píxel de 20x20m y que coincida con los pixeles de 10x10m.

Partiendo de los puntos generados a 20x20m, se aplica un buffer de 5,65m, para así seleccionar únicamente los puntos de 10x10m que se encuentren dentro de ese buffer, es decir, un único punto. De este modo se consigue una correspondencia uno a uno entre puntos obtenidos para un paso de 20x20m y los obtenidos para 10x10m.

De otro modo, dentro de los píxeles de 20x20m, habría 4 puntos de 10x10m (Figura 5), lo que daría lugar a un muestreo con valores repetidos, es decir, 4 puntos de muestreo tendrían el mismo valor (de las imágenes Sentinel-1 y Sentinel-2) al estar dentro de un mismo píxel. Un buffer a una distancia de 5,65m permite solucionar este problema, dado que sólo uno de los cuatro puntos cae en su interior (Figura 5).



*Figura 5: Muestra puntos: los puntos morados corresponden a los píxeles de 20x20 y los puntos amarillos corresponden a los píxeles de 10x10m*

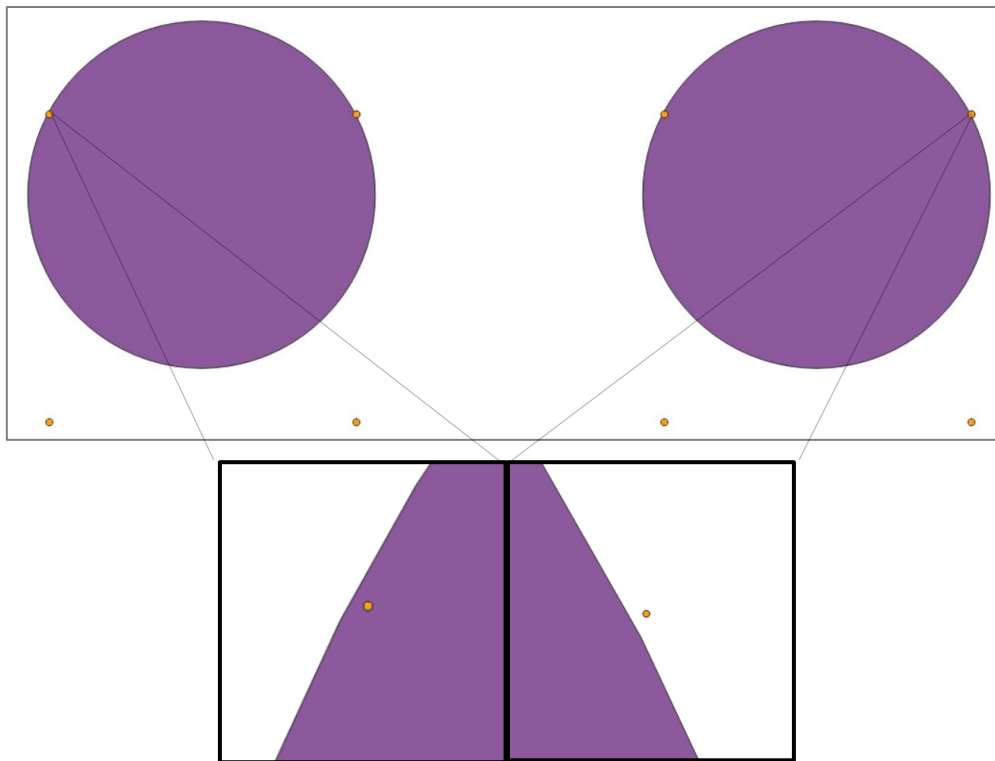


Figura 6 : Muestra de cómo el buffer generado solo cubre un único punto de los cuatro:

Finalmente, se seleccionan por localización los puntos que se encuentran dentro del buffer (Figura 6) y se obtienen los puntos para el muestreo (ver código en el ANEXO VI:).

#### 3.4.5.1. Selección del conjunto de datos de trabajo

A partir de la malla de puntos generada en el paso anterior, se realiza un muestreo de los puntos que constituirán el conjunto de datos de trabajo. Estos puntos se utilizarán para extraer las métricas LiDAR y datos Sentinel-1 y Sentinel-2, así como la información de las variables forestales medidas en campo. Este conjunto de datos constituirá la base para el ajuste de los modelos de regresión.

En primer lugar, el trabajo se restringe a dos especies arbóreas por acotar el alcance de este TFM. Entre las especies disponibles, se selecciona el haya, *Fagus sylvatica*, y el pino silvestre, *Pinus sylvestris*, por ser las especies con más representación en Navarra, más del 45% y 15% respectivamente, y ser una de ellas frondosa y la otra conífera.

En lugar de trabajar con todos los puntos, se realiza un muestreo del 5%, que para que sea equilibrado entre los distintos rangos de valores de las variables forestales de interés, se opta por un muestreo aleatorio estratificado. Así se obtiene un muestreo de puntos equilibrado para todo el rango de datos. En el script del ANEXO VI: se detalla cómo se ha realizado el muestreo.

En la (Figura 7) se muestran los recintos de haya y los puntos seleccionados mediante el muestreo aleatorio estratificado (se debe tener en cuenta que la escala de representación del tamaño de los puntos de muestreo se ha exagerado para facilitar la visualización):



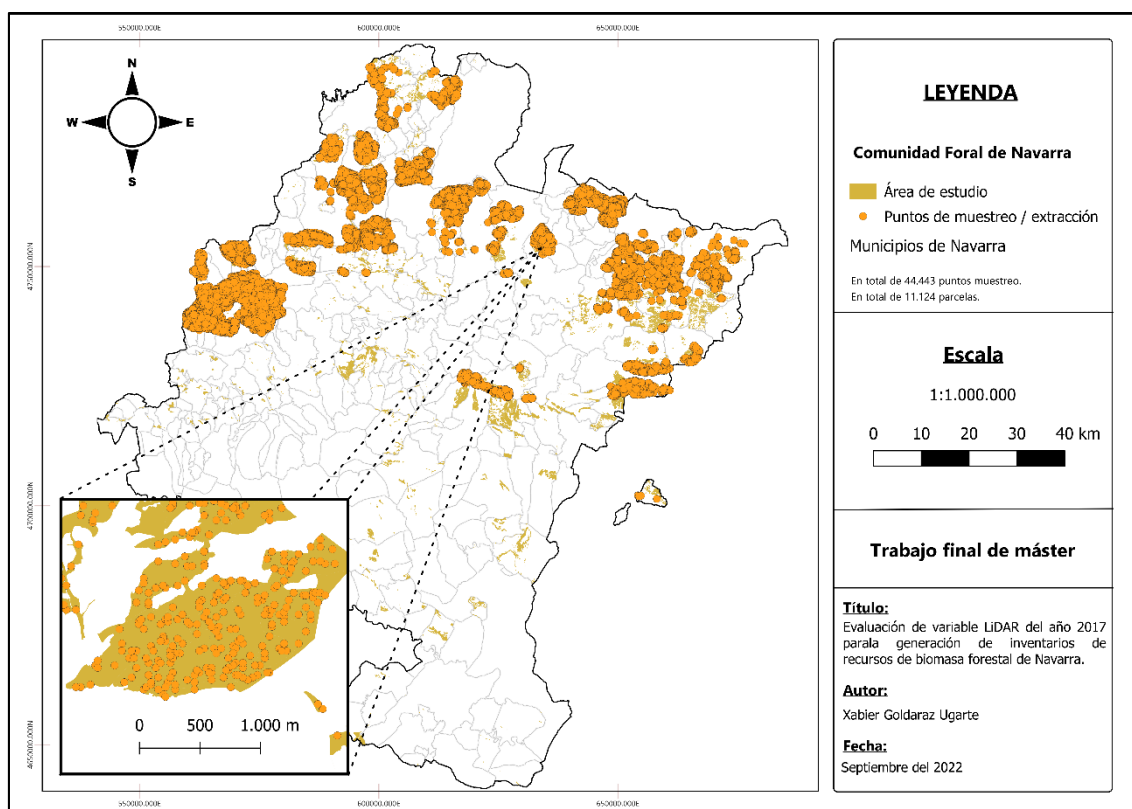


Figura 7: Muestreo realizado para *F. sylvatica*

### 3.4.6. EXTRACCIÓN DE MÉTRICAS

La extracción de las métricas LiDAR para los puntos de muestreo se ha realizado mediante un código de Python (ANEXO III:), que a partir de las coordenadas X e Y de cada punto de muestreo busca el dato correspondiente a su píxel en los mosaicos de las métricas LiDAR que queda almacenado en un data frame y finalmente se exporta en formato xlsx.

### 3.4.7. ANÁLISIS DE CORRELACIÓN

Con los datos extraídos se realizó un análisis de correlación entre las métricas LiDAR con el objetivo de identificar las que aportan una información similar o redundante y las que ofrecen información complementaria. De forma previa se observó que algunas de las métricas obtenidas no seguían una distribución normal, por lo que se optó por calcular tres matrices de correlación utilizando tanto el coeficiente de correlación Pearson (que asume linealidad), como los de Spearman y el de Tau-b de Kendall, estos dos últimos son de gran utilidad en aquellos casos donde las variables de estudio no siguen un comportamiento normal (Fassnacht et al., 2021).

### 3.4.8. CÁLCULO DE MODELOS DE REGRESIÓN

Finalmente, se llega al último apartado de la metodología que consiste en el cálculo de modelos de regresión que permitan estimar las variables forestales AD, AB y VT a partir de las métricas LiDAR y datos Sentinel-1 y Sentinel-2 mencionados. Para ello se utilizará el algoritmo de aprendizaje automático Random Forests (RF) (Breiman, 2001), algoritmo no paramétrico de gran flexibilidad que ha proporcionado buenos resultados en estudios similares (Yu et al., 2010) (Huang et al., 2022).



Desde su creación hace dos décadas RF ha ganado popularidad rápidamente y a día de hoy, es usado de forma masiva en numerosos ámbitos y entre ellos el de la teledetección (Gigović et al., 2019). Técnicamente es un algoritmo de aprendizaje conjunto que utiliza la técnica de embolsado (bagging) para generar un número elevado de estimadores (árboles de decisión) en subconjuntos de datos (tanto en bolsas de observaciones como en subconjuntos de variables) (Pal, 2005), lo que asegura la diversidad en los estimadores y evita los problemas de sobre-entrenamiento de otras técnicas basadas en árboles de decisión (Li et al., 2019). RF es además muy eficiente desde un punto de vista computacional y su entrenamiento es rápido, no requiere de un elevado número de parámetros de ajuste y sus resultados son menos sensibles a esos parámetros de ajuste que los de otros algoritmos de aprendizaje automático (Li et al., 2019).

Utilizar este tipo de algoritmos con grandes conjuntos de datos, como en este trabajo, tiene claras ventajas, ya que a pesar de tener un peso de datos muy alto suele producir modelos precisos que se ejecutan de manera muy eficiente (Belgiu & Drăgu, 2016).

Para ajustar los modelos RF, en primer lugar (Fase 1), se han entrenado modelos utilizando un subconjunto de datos de entrenamiento (70% de la muestra) probando diferentes combinaciones de parámetros y evaluando su rendimiento a lo hora de predecir ese mismo conjunto de datos de entrenamiento (obteniendo métricas de calidad como  $R^2$  o RMSE). Una vez realizado el entrenamiento del algoritmo, con el conjunto de datos restante (30% en este caso) se realiza la validación (Fase2), calculando nuevamente las métricas de calidad, pero en esta ocasión con el conjunto de datos de validación o test (ANEXO V:).

El ajuste de parámetros se realiza de forma empírica, ejecutando un elevado número de modelos con combinaciones diferentes de parámetros para evaluar el rendimiento de cada modelo (tanto sobre los datos de entrenamiento como validación). Para ello, se ha utilizado el método GridsearchCV, que encuentra los parámetros de mejor rendimiento entre todas las posibilidades de combinaciones, método que resulta más efectivo cuando el número de combinaciones no es excesivo (Li et al., 2019).

La selección de los parámetros a optimizar se ha basado en la bibliografía (Torre-Tojal et al., 2022), siendo los parámetros seleccionados los siguientes:

- N\_estimators
- Max\_features
- Max\_depth
- Min\_samples\_split

A continuación, se describen los detalles de cada parámetro (Saxena, 2020).

#### *3.4.8.1. n\_estimators: Número de árboles (10,50,100)*

A priori una mayor cantidad de árboles debería de producir un resultado mejor, aunque tradicionalmente se observa cierta tendencia a la saturación, por lo que a partir de un determinado valor el rendimiento del modelo se estabiliza (Belgiu & Drăgu, 2016). Por otro lado, incrementos en el valor de este parámetro implican un aumento considerable en las necesidades de cálculo y, por ende, en el tiempo de cómputo.

En la (Figura 8) se aprecia que a valores bajos de n\_estimators el rendimiento del modelo aumenta rápidamente y luego se estanca a partir de 50-100. Por tanto, no es conveniente

elegir un valor excesivo de este parámetro, ya que no proporcionará rendimientos mayores e implicará un coste computacional excesivo.

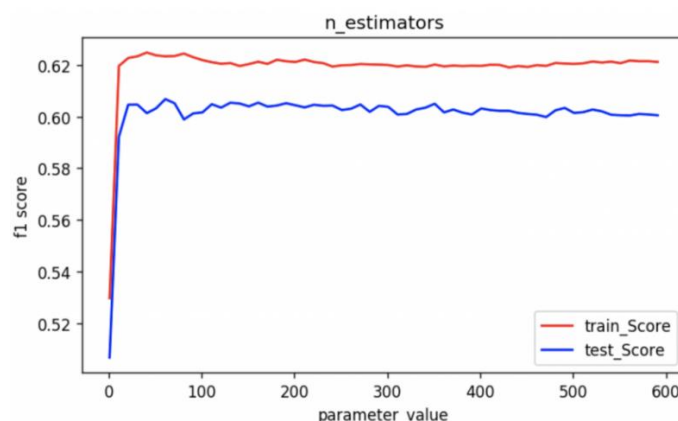


Figura 8: Ejemplo de  $n\_estimators$  (Saxena, 2020)

### 3.4.8.2. $max\_features$ : (1, sqrt, log2, None)

Este parámetro establece el número de variables predictivas que son evaluadas en los nodos de cada árbol para dividir el conjunto de datos hacia una y otra rama.

Es un parámetro que afecta a la diversidad en los árboles que componen el RF. Un número reducido hará que cada árbol sea distinto, pero probablemente producirá un rendimiento bajo (habrá árboles a los que les toquen variables con poca capacidad predictiva y por tanto fallen). Por el contrario, un valor elevado de  $max\_features$  permitirá que cada árbol se construya con prácticamente todo el conjunto de variables predictivas, resultando en que todos los árboles del RF sean prácticamente idénticos (todos seleccionarán las mejores variables). Por tanto, es probable que un número elevado de este parámetro produzca un sobreajuste a los datos de entrenamiento utilizados para construir el RF.

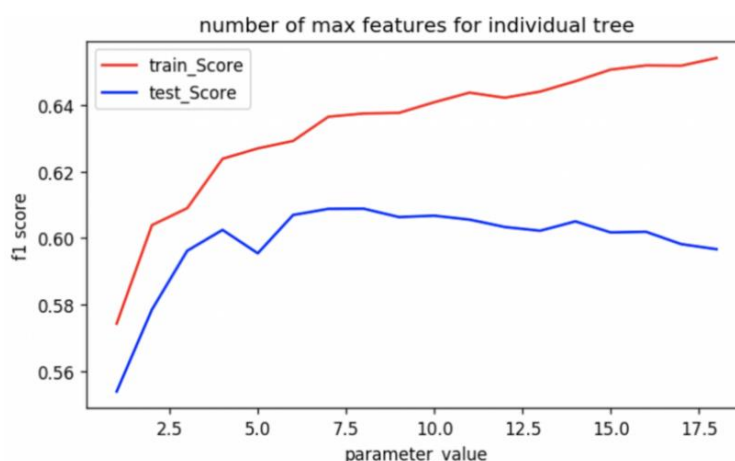


Figura 9: Ejemplo de  $max\_features$

En la (Figura 9) se aprecia que el rendimiento en entrenamiento y en validación dibujan curvas diferentes. A valores altos se tiende al sobreajuste (el rendimiento en entrenamiento sigue subiendo, pero en validación baja). Por tanto, es necesario seleccionar un valor intermedio que suele rondar la raíz cuadrada del número total de variables predictivas utilizada (Belgiu & Drăgu, 2016).

### 3.4.8.3. *max\_depth: Profundidad máxima (5, 10, None)*

La profundidad máxima de los árboles del RF se define como la ruta más larga entre el nodo raíz y el nodo hoja. Por tanto, el parámetro `max_depth` puede limitar la profundidad hasta la que se deja crecer cada árbol.

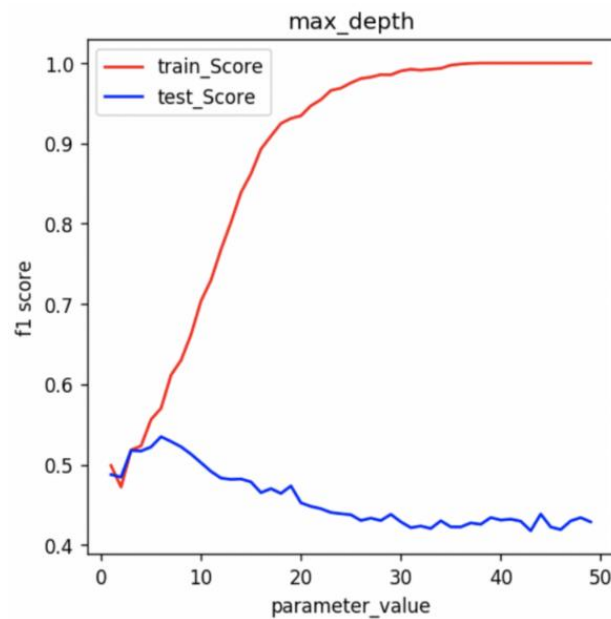


Figura 10: Ejemplo de `max_depth`

En la (Figura 10), se observa que valores altos de este parámetro también tienden al sobreajuste. Árboles muy profundos permiten reproducir de forma casi perfecta el conjunto de entrenamiento, pero fallan al aplicarse a los datos de validación (no vistos en el entrenamiento) por lo que tienen una baja capacidad de generalización.

### 3.4.8.4. *min\_samples\_split: Número mínimo de observaciones (2, 10, 20)*

Este parámetro indica el número mínimo de observaciones en un nodo para dividirlo. Valores altos generan nodos terminales (hojas) con muestras variadas (algunas de ellas erróneas), mientras que valores bajos permiten dividir sucesivamente el nodo y afinar las predicciones. Por ejemplo, un valor de 2 indica que, si algún nodo terminal tiene más de dos observaciones y no es un nodo puro, se dividirá aún más en dos subnodos.

### 3.4.8.5. *Calidad del modelo:*

La calidad de los modelos, la cual determina estadísticamente el ajuste del modelo, se ha calculado de forma cuantitativa, tanto para los datos de entrenamiento como para los datos de test. Para ello, se han utilizado el  $R^2$  y el RMSE:

Coeficiente de determinación  $R^2$ : Representa la proporción de variación de la variable respuesta que es explicada por las variables predictivas. Cuanto más cercano a 1 mejor. De hecho, si su valor fuera 1 las variables predictivas explicarían toda variación de la variable respuesta, y la predicción no tendría error.

Raíz del Error Cuadrático Medio (RMSE): Este valor indica el error del modelo, por tanto, cuanto mayor sea su valor peor será el ajuste del modelo. A diferencia de otras métricas de error el RMSE mantiene las unidades de la variable respuesta.

### 3.4.8.6. Extracción de datos Sentinel 1 y Sentinel 2:

Para realizar la comparación entre los modelos de regresión basados en métricas LiDAR con aquellos basados en Sentinel-1 y Sentinel-2, se utilizaron imágenes satelitales de estos satélites adquiridas en el año 2018. Los datos de Sentinel 1 corresponden a las medias de las bandas de cada mes para el año 2018 de las pasadas ascendentes. Por otro lado, los datos de Sentinel 2 corresponden a las medias mensuales del 2018. Para que estos datos pudieran ser procesados de forma similar a las métricas LiDAR, se realizaron compuestos anuales para cada banda. Estos compuestos se calcularon mediante el script del ANEXO VII.:

Así, se obtuvo para cada satélite las siguientes bandas (Tabla 9)

Tabla 9: Bandas de los Satélites Sentinel-1 y Sentinel-2 utilizadas.

SATÉLITE	BANDA
<b>Sentinel1</b>	Retrodispersión VH
	Retrodispersión VV
	Ratio VH/VV
<b>Sentinel2</b>	B02
	B03
	B04
	B05
	B06
	B07
	B08
	B8A
	B11
	B12
	NBR (índice de vegetación)
	NDII (Índice de vegetación)
	NDVI (Índice de vegetación)

Por otro lado, de la misma manera que se han calculado las extracciones de los datos LiDAR, para los compuestos de Sentinel-1 y Sentinel-2 se ha utilizado el mismo script del ANEXO V.:

### 3.4.8.7. Modelos de regresión ajustados:

Se han ajustado modelos de regresión utilizando distintos tipos de datos de entrada: LiDAR, Sentinel-1 y/o Sentinel-2, que se combinan entre sí tal y como se indica a continuación:

- Modelo 1: LiDAR
- Modelo 2: Sentinel-1
- Modelo 3: Sentinel-2
- Modelo 4: LiDAR + Sentinel-1
- Modelo 5: LiDAR + Sentinel-2

En definitiva, se han ajustado 5 modelos de regresión para cada una de las tres variables forestales de interés (altura dominante, área basimétrica y volumen total), otorgando un total de 15 modelos diferentes.

### 3.5. DESCRIPCIÓN DE HERRAMIENTAS UTILIZADAS

Para realizar el proceso del trabajo se han utilizado diferentes herramientas. Las principales utilizadas han sido el intérprete de Python Spyder (anaconda), QGIS y LAStools.

#### 3.5.1. QGIS VERSIÓN 3.22.7

La herramienta de QGIS ha servido para realizar el proceso de muestreo, 3.4.5, así como para la composición de los mapas de los mosaicos LiDAR y diversos cálculos de métricas como el cálculo del coeficiente de variación de la intensidad y la elevación.

#### 3.5.2. PYTHON 3.9 Y SPYDER ANACONDA:

Python es un lenguaje de programación interpretado y de código abierto que se utiliza para desarrollar aplicaciones de todo tipo. En este trabajo Python, y en concreto su intérprete Spyder (Anaconda), ha sido el entorno de desarrollo y automatización de los procesos llevados a cabo. Además de los módulos básicos instalados por defecto en Spyder, se han instalado desde el siguiente repositorio de módulos: "Archived Unofficial Windows Binaries for Python Extension Packages" (<https://www.lfd.uci.edu/~gohlke/pythonlibs/>), los siguientes módulos específicos:

- Fiona
- Rasterio
- Pyproj
- Gdal
- Shapely
- Geopandas

#### 3.5.3. LAStools:

Existen diferentes herramientas, tanto comerciales como libres, que permiten procesar nubes de puntos LiDAR. En este caso se han utilizado las herramientas de la suite LAStools, que es una colección de herramientas programables que permite procesar archivos de múltiples formatos (LAS, LAZ comprimido, Terrasolid BIN, ESRI Shapefiles, ASCII etc), realizar correcciones y transformaciones muy diversas y generar productos a modo de imágenes ráster (MDT, MDS, imágenes de intensidad, etc.).

La Tabla 10 lista las herramientas de LAStools utilizadas.

Tabla 10: Herramientas de LAsTools utilizadas

HERRAMIENTAS	DESCRIPCIÓN
<b>lastile</b>	Divide archivos LAS/LAZ en teselas más pequeñas para facilitar su procesamiento. Opcionalmente, la herramienta también puede crear un pequeño buffer alrededor de cada tesela.
<b>lasgrid</b>	Lee una nube de puntos y genera un archivo ráster de salida con un tamaño de píxel dado. La herramienta puede rasterizar la elevación o la intensidad y asignar a cada píxel de salida el dato del punto más bajo o el más alto, o calcular estadísticos como la altura promedio o la desviación estándar.
<b>lascanopy</b>	Realiza cálculos de utilidad forestal partir de nubes de puntos LiDAR de altura normalizada y genera archivos ráster de salida como la densidad del dosel, la fracción de cabida cubierta, los percentiles de altura o intensidad, promedios, mínimos, máximos, curtosis, asimetría, desviación estándar, etc..
<b>las2dem</b>	Es una herramienta que triangula puntos LIDAR en un TIN temporal y luego rasteriza el TIN para crear un DEM. La herramienta puede rasterizar elevación, pendiente o intensidad.
<b>lasnoise</b>	Marca o elimina puntos identificados como ruido en nubes de puntos.
<b>lasinfo</b>	Es una herramienta útil para informar sobre el contenido del encabezado de los archivos. Imprime un breve resumen con la descripción general del contenido de un archivo LAS/LAZ, estadísticos descriptivos, etc.
<b>las2las</b>	Herramientas que lee un archivo LIDAR y lo guarda como otro archivo tras aplicar operaciones como filtrar, transformar, proyectar, diluir o modificar su contenido
<b>lasheight</b>	Esta herramienta calcula la altura de cada punto sobre el suelo, es decir, la altura normalizada. Adicionalmente, también puede convertir alturas de ortométricas a elipsoidales, aunque para ello se necesita el modelo de Geoide a aplicar.
<b>lasmerge</b>	Esta herramienta sirve para fusionar varios archivos LAS/LAZ en uno.

## 4. RESULTADOS Y DISCUSIÓN

En este apartado se muestran los resultados obtenidos en este TFM. Está dividido en 4 apartados: el primero, corresponde a un análisis descriptivo previo de los datos de las ordenaciones forestales. Se muestra por otra parte, las correlaciones entre las métricas LiDAR con tres coeficientes de correlación diferentes. En el tercero apartado se muestra un ejemplo de los mosaicos obtenidos para todo el territorio a raíz del cálculo de las métricas. Y finalmente, en el último apartado, se muestran los resultados de los modelos de regresión para *F.sylvatica* y *P.sylvestris*.

Los resultados obtenidos a lo largo del proyecto han ido condicionando el desarrollo del estudio sobre todo aquellos procesos que impedían procesar los datos debido al volumen de estos y a la necesidad de tiempo de procesamiento de los mismos. Debido a estos problemas encontrados, la metodología que se propuso en un principio fue variando, ajustándose a las necesidades de procesado y evaluación de los resultados consiguiendo finalmente los resultados que se muestran a continuación.

### 4.1. ANÁLISIS DESRIPTIVO DE LOS DATOS DE CAMPO:

Primeramente, mostramos en la siguiente Tabla 11, los estadísticos descriptivos de cada una de las variables a estimar para cada especie.

Tabla 11: Estadística descriptiva de los datos de las ordenaciones forestales

	nombre	área	% área sobre Total	
	Haya	359,04	45,79	
	Pino silvestre	123,01	15,69	

ALTURA DOMINANTE							
nº polígonos	mean	std	min	25%	50%	75%	max
3346	25,28	10,29	0,92	19,00	25,00	30,74	103,40
3208	32,13	15,83	0,92	21,00	31,60	43,00	94,00

AREA BASIMÉTRICA							
nº polígonos	mean	std	min	25%	50%	75%	max
3346	21,16	5,26	3,00	18,00	21,94	25,00	43,00
3208	16,66	7,55	2,00	12,60	16,00	20,00	191,00

VOLUMEN							
nº polígonos	mean	std	min	25%	50%	75%	max
3346	817,88	8635,66	0,01	136,08	215,01	284,00	286434,66
3208	233,04	183,24	2,68	114,72	197,84	309,96	2425,60

Existen en total 3.346 polígonos de las ordenaciones forestales que corresponden a masas de la especie *F.sylvatica* y 3.208 polígonos para la especie de *P.sylvestris*. Estos van a ser entonces, el punto de partida del trabajo.

Cada especie tiene una representación diferente frente al conjunto de masas facilitadas. Por un lado, el haya, cuenta con un área total de 359,04 km<sup>2</sup> que corresponde a un 45,8% del total, es decir, casi un 50% de las masas corresponden a hayedos. Por otro lado, el

pino silvestre, en cambio, tiene un área total de 123,01 km<sup>2</sup> y representa cerca del 16% sobre el total. Estas dos especies son las dos especies con más representación de todas las especies de las ordenaciones forestales facilitadas.

En cuanto a las variables de interés, los valores más altos de altura basimétrica los tiene el haya, pero en cambio, la media más alta la tiene el pino. No son grandes diferencias, apenas el pino supera en 7m al haya. Por otra parte, en el área basimétrica se puede encontrar alguna diferencia más considerable. El dato máximo de área basimétrica para el pino es de 191m<sup>2</sup>/ha frente a 43m<sup>2</sup>/ha que contiene el haya. A pesar de ello, la media para las dos especies es muy parecida, 21m<sup>2</sup>/ha para el haya y 16m<sup>2</sup>/ha para el pino. Finalmente, en la última variable, el volumen, sí que se puede apreciar diferencias tanto en los valores máximos como en la media. Obteniendo una diferencia de 600m<sup>3</sup>/ha. Como dato impactante, cabe mencionar el valor máximo del haya, 286.434m<sup>3</sup>/ha, que probablemente se deba a un error, ya que es 2000 veces la media.

Por otro lado, se muestran en la siguiente figura, los histogramas de las variables de interés para el haya (Figura 11):

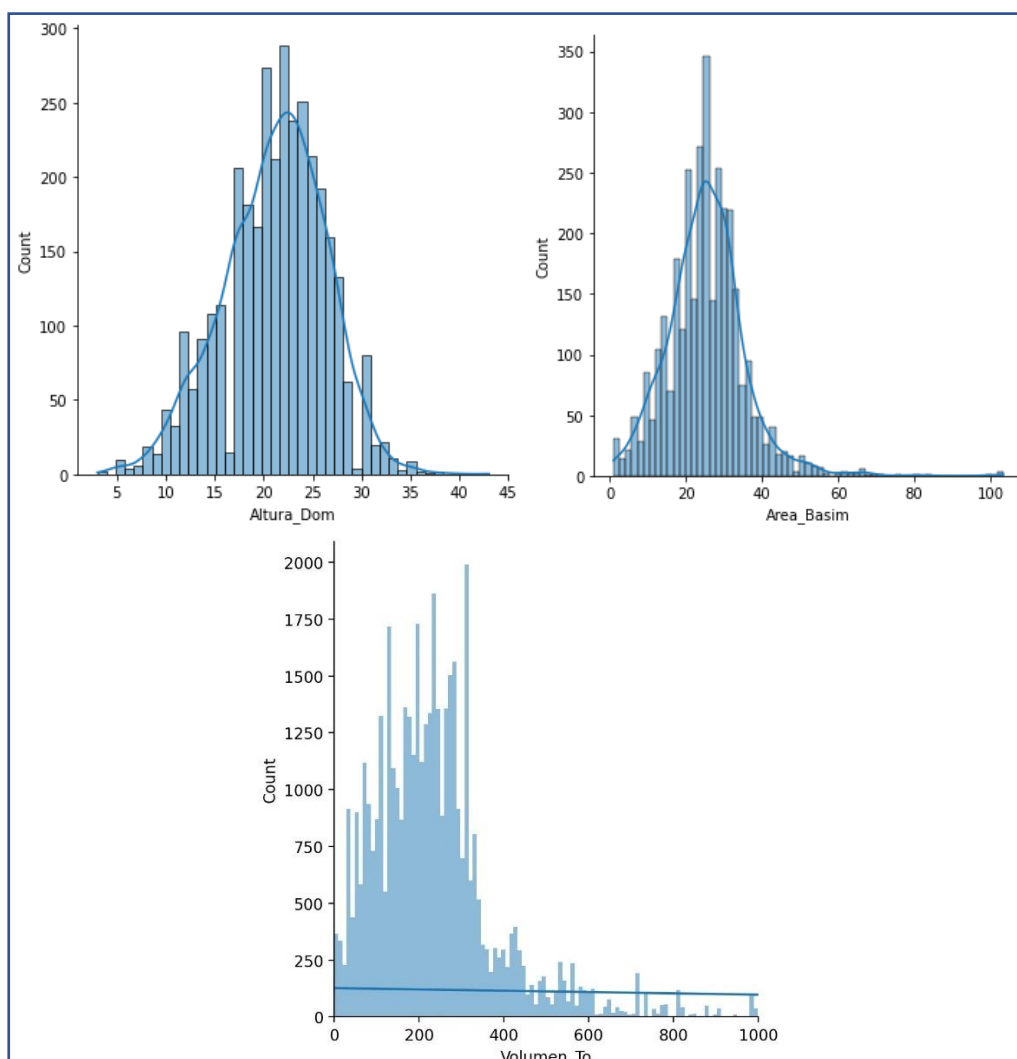


Figura 11: Histogramas de las variables de interés de *F.sylvatica* respecto a los datos de ordenaciones forestales



A primera vista, se puede pensar que los datos de estas variables siguen una distribución normal, dado su figura del histograma. Se han representados los valores más frecuentes, dejando los valores anómalos a un lado.

Finalmente, también se representan los histogramas del pino para las tres variables (Figura 12):

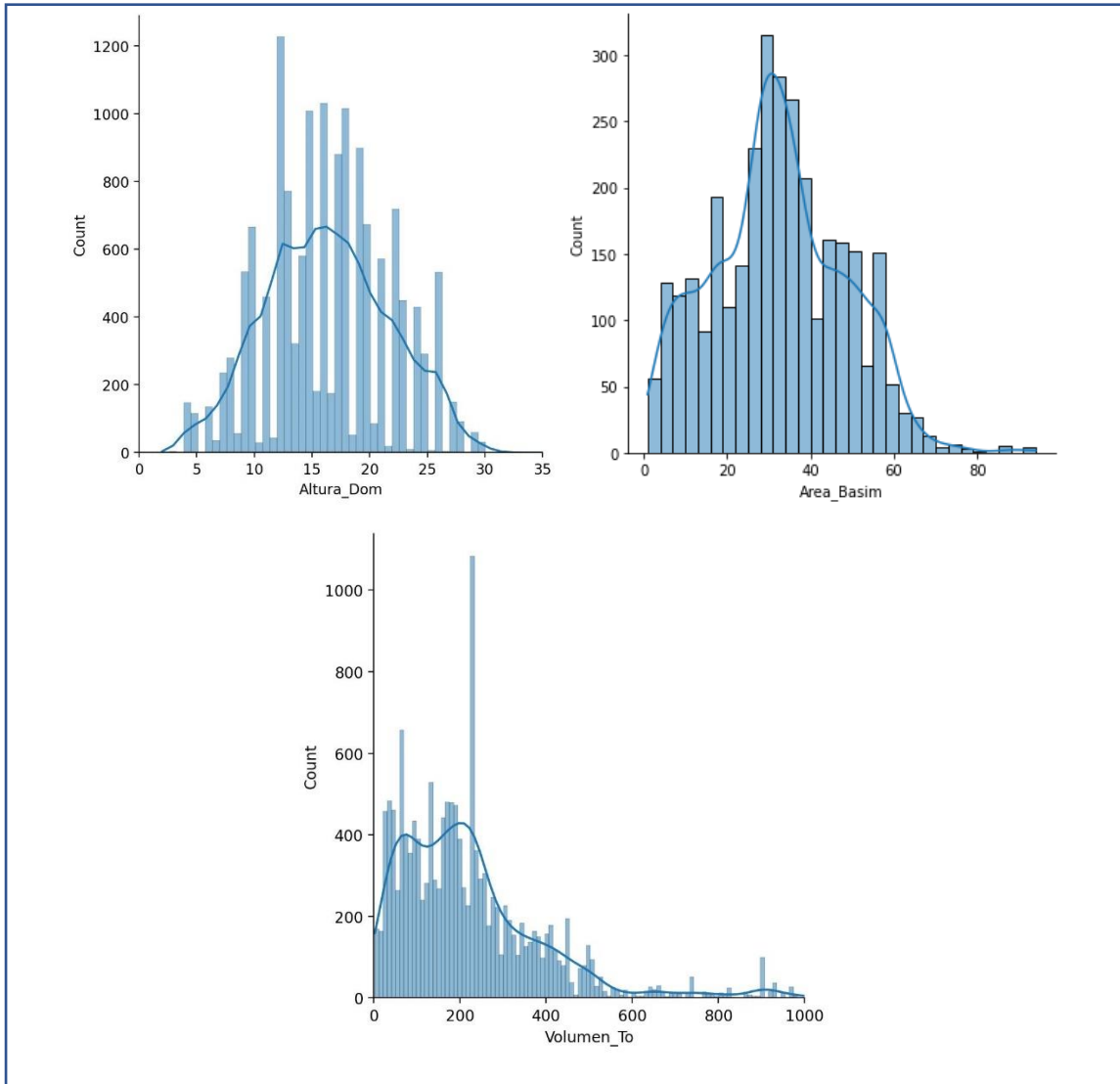


Figura 12: Histogramas de las variables de interés de *P.sylvestris* respecto a los datos de ordenaciones forestales

En el caso del pino, también parece deducirse que las variables siguen una distribución normal, excepto, en el caso del volumen, donde tiene una tendencia más lineal.

#### 4.2. MAPAS:

En este apartado se quiere mostrar un ejemplo de los mosaicos creados para todas las métricas LiDAR. El resto de los mosaicos están adjuntos en el ANEXO IX:

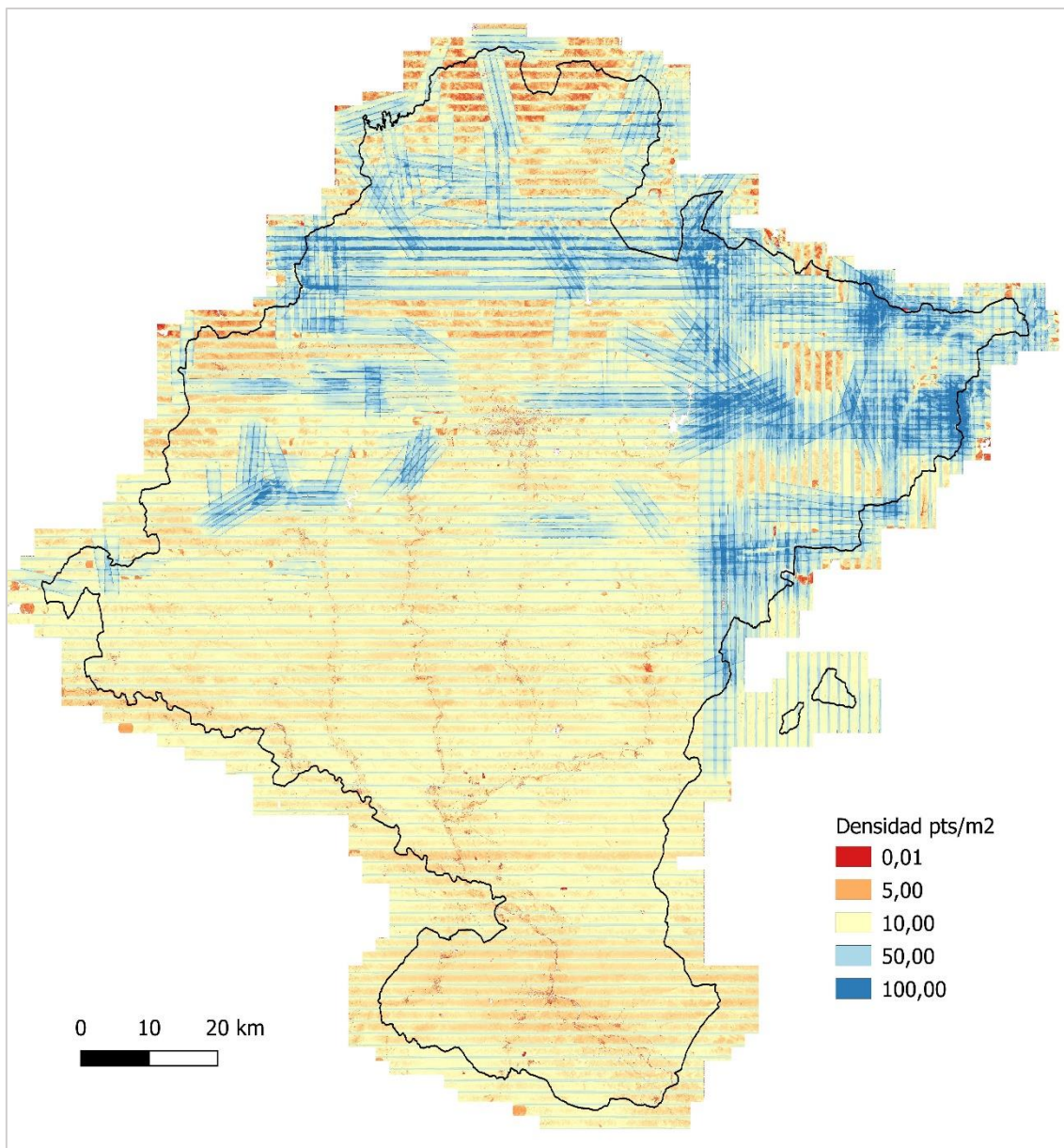


Figura 13: Mosaico de la densidad LiDAR.

En la (Figura 13) se puede observar el mosaico de densidad de puntos por m<sup>2</sup> para todo el territorio de Navarra. En los detalles de la segunda cobertura LiDAR, del año 2017, se informa que en este vuelo hay una densidad media de 14pts/m<sup>2</sup>. En el siguiente apartado, 4.2.1, se muestra la densidad media de las especies forestales estudiadas y se compara con ese valor medio del vuelo. A pesar de ello, esta imagen muestra una clara diferencia de densidad entre las zonas norte y sur de Navarra, que se corresponden con zonas abruptas y zonas llanas, respectivamente. Esto se explica, ya que es necesario más pasadas para poder caracterizar adecuadamente el terreno abrupto, debido a los desniveles existentes y la altura de vuelo que se fija. Por otra parte, se pueden observar líneas paralelas de color azul claro dirección E-O. Esto se debe al solape entre pasadas que genera una mayor densidad de puntos en determinadas zonas.

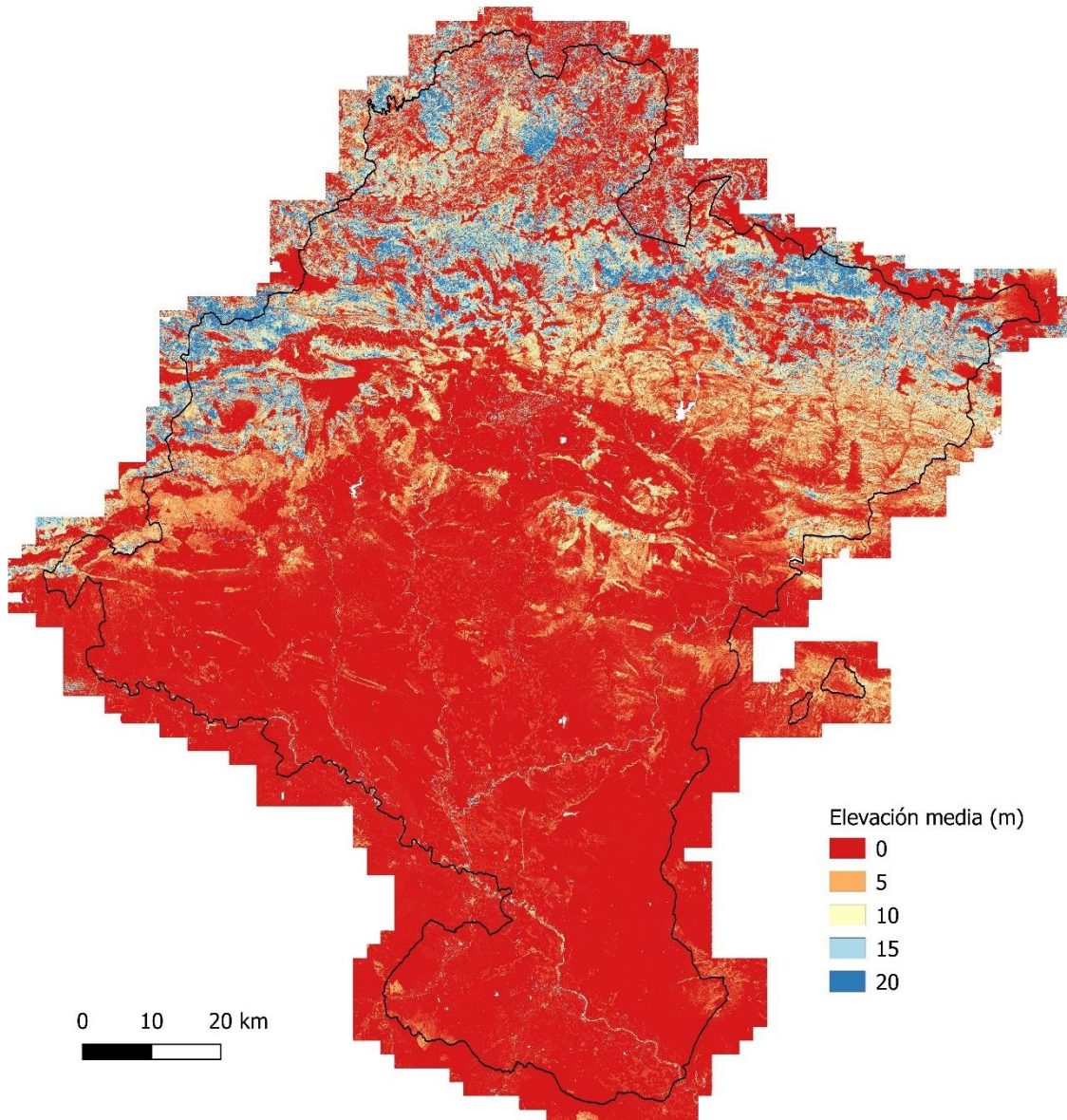


Figura 14: Mosaico de la Elevación media (m) LiDAR

En la (Figura 14) se representa la métrica de elevación media obtenida. Los valores más rojizos representan una altura más baja y los valores más azulados, en cambio, una altura más elevada. Son las zonas de la mitad norte las que tienen una elevación más alta, ya que aquí se encuentran los grandes bosques de Navarra. Por otra parte, también destacan las siluetas de los ríos en la parte sur, ya que a los lados por donde discurren los ríos Arga, Ega y Aragón se encuentran bosques de ribera con una altitud considerable.

#### 4.2.1. ESTADÍSTICA DESCRIPTIVA:

Una vez creado los mosaicos, se pueden obtener estadísticos descriptivos de las métricas LiDAR. Así pues, en este apartado se visualizan esos estadísticos descritos en el apartado de metodología para las dos especies forestales.

Primeramente, se muestran los datos para el haya y después para el pino.



Tabla 12: Métrica densidad y métricas LiDAR relacionadas con la elevación para la especie *F.sylvatica*, parte 1.

	density	Elevation _average	Elevation _std	Elevation _cv	Elevation _lowest
<b>count</b>	44443	44443	44443	44443	44443
<b>mean</b>	33,71	12,92	5,37	-	0,92
<b>std</b>	31,31	6,46	2,60	-	3,46
<b>min</b>	0,41	0,00	0,00	-	0
<b>25%</b>	10,15	8,12	3,34	0,27	0,00
<b>50%</b>	25,25	12,77	5,21	0,48	0,00
<b>75%</b>	44,78	17,56	7,13	0,73	0,00
<b>max</b>	301,03	48,57	17,31	35,50	41,46

Tabla 13: Métricas LiDAR relacionadas con la elevación para la especie *F.sylvatica*, parte 2.

	Elevation _p25	Elevation _p50	Elevation _p75	Elevation _p90	Elevation _p95	Elevation _highest	Fcc_3m
<b>count</b>	44443	44443	44443	44443	44443	44443	44443
<b>mean</b>	9,48	13,88	16,79	18,59	19,41	21,57	0,69
<b>std</b>	8,16	7,61	7,01	6,65	6,49	6,26	0,20
<b>min</b>	0,00	0,00	0,00	0,00	0,00	0	0
<b>25%</b>	0,34	8,48	12,47	14,57	15,50	17,74	0,60
<b>50%</b>	9,19	14,64	17,43	19,12	19,86	21,92	0,75
<b>75%</b>	16,24	19,55	21,77	23,23	23,91	25,82	0,85
<b>max</b>	46,90	49,15	50,65	51,80	52,46	110,14	1,12

En estas dos primeras tablas, Tabla 12 y Tabla 13, se puede observar los valores de las elevaciones para el haya, que tiene de elevación máxima media, 21,6m. La densidad de puntos media cuenta con 33 pts/m<sup>2</sup>, el doble que la media para todos los datos de la segunda cobertura LiDAR. Esto se debe, a que la distribución del haya, como se puede observar en el apartado de la metodología 3.4.5 en la Figura 7, está limitado a las zonas norte, donde la densidad de puntos como se ha explicado en el apartado 4.2 es mucho mayor.

Por otra parte, hay que señalar el número de puntos total para el haya, que es de 44.443. Como se podrá ver posteriormente, el pino cuenta con menos puntos de muestreo, ya que también representaba menos área de masa forestal.

Por comentar los valores que son más distintivos, se puede apreciar como los valores mínimos de prácticamente todas las métricas de elevación son 0 o cercanas a 0. Si comparamos estos valores con la elevación mínima, se puede comprobar como la media de la elevación mínima es muy cercana al 0. También hay que destacar el valor máximo dentro de esta métrica, ya que es un dato bastante peculiar para la métrica la cual corresponde a 41,46m. Esto se explica porque se están muestreando píxeles de 10x10m, por lo que puede ser que un punto de muestreo haya coincidido en un píxel con una masa arbolada muy alta en la que la altura mínima es de ese valor.

Tabla 14: Métricas LiDAR relacionadas con la intensidad para la especie *F.sylvatica*.

	intensity_average	intensity_cv	intensity_highest	intensity_lowest	intensity_std
<b>count</b>	44443	44443	44443	44443	44443
<b>mean</b>	5587,36	-1,14849E+35	7045,80	4802,28	415,11
<b>std</b>	473,50	6,2505E+36	1065,88	229,00	267,31
<b>min</b>	4837,00	-3,40282E+38	5200,00	4169,00	0,00
<b>25%</b>	5330,00	0,04	6279,00	4664,00	241,00
<b>50%</b>	5477,00	0,06	6809,00	4794,00	328,00
<b>75%</b>	5672,00	0,09	7554,00	4922,00	500,00
<b>max</b>	11102,00	0,36	14534,00	9552,00	2771,00

En esta Tabla 14, se muestran los valores de intensidades para el haya, que oscilan entre los 4.800 y 11.000 de intensidad media. Datos que se corresponden con el número de bits del vuelo, 16 bits, así, pues, los datos están dentro de 1 y 65.536.

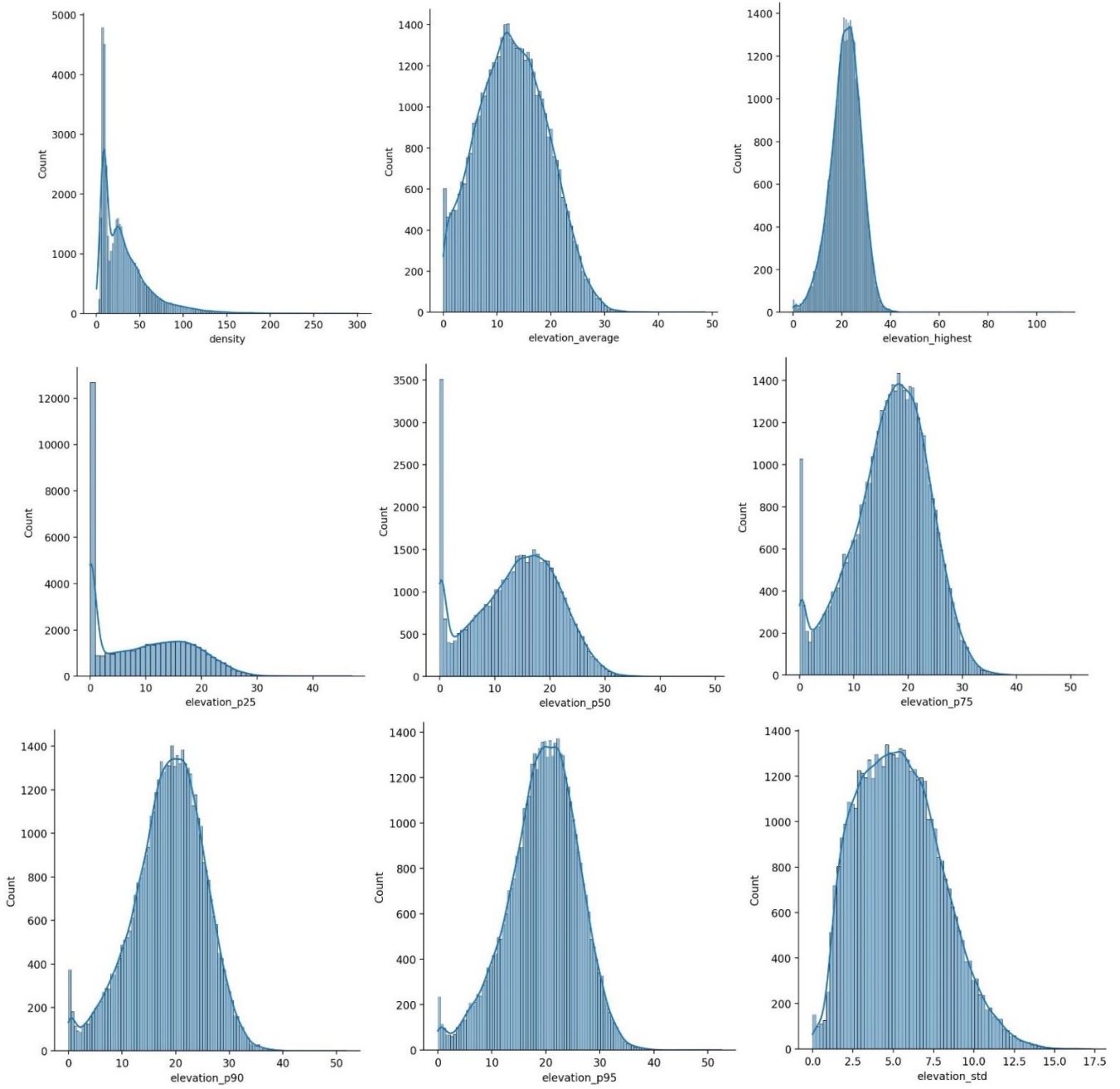
Los valores de intensidad varían principalmente según el color de la superficie del objeto donde el láser LiDAR se refleja, en este caso, valores bajos de intensidad corresponderán a materiales oscuros y, por el contrario, valores altos de densidad corresponderán a objetos más claros. Por otro lado, hay que subrayar que la intensidad no es una magnitud que esté bien calibrada, por lo tanto, la interpretación de estos datos se debe hacer con cautela.

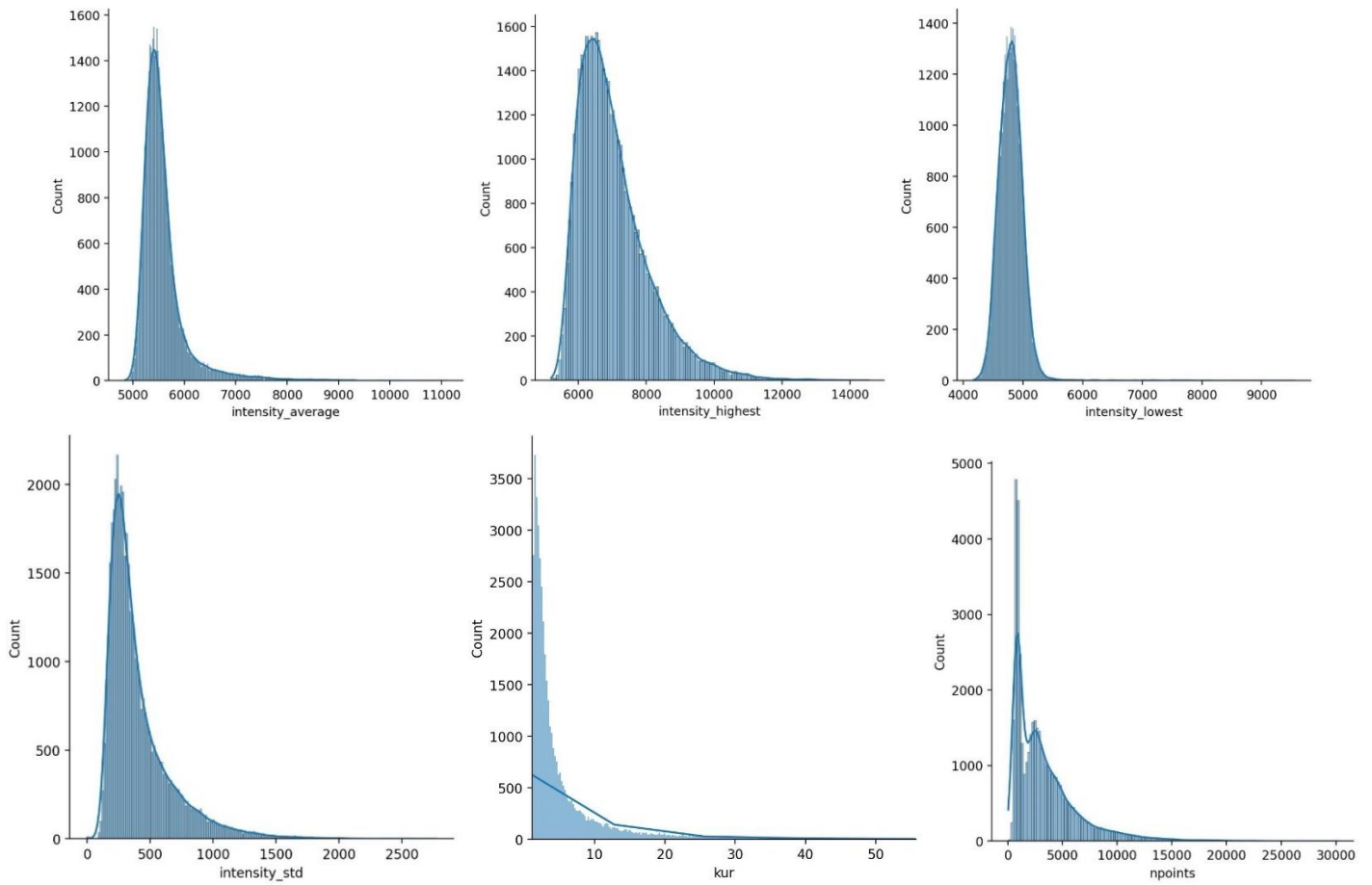
 Tabla 15: Métricas LiDAR relacionadas con la forma para la especie *F.sylvatica*.

	kur	npoints	Range c00	Range c01	Range c02	Range c03	Range c04	Range c05	Range c06	ske
<b>count</b>	44443	44443	44443	44443	44443	44443	44443	44443	44443	44443
<b>mean</b>	7,22	3371,46	760,55	476,54	612,25	759,56	539,20	191,23	29,23	-0,83
<b>std</b>	28,89	3130,59	1124,33	905,75	997,49	1170,76	1041,28	624,38	217,54	1,85
<b>min</b>	0,00	41,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	-13,82
<b>25%</b>	1,93	1015,00	65,00	8,00	29,00	33,00	0,00	0,00	0,00	-1,61
<b>50%</b>	3,01	2525,00	346,00	123,00	225,00	318,00	70,00	0,00	0,00	-0,73
<b>75%</b>	6,11	4477,50	998,00	530,50	759,00	991,00	634,00	18,00	0,00	-0,04
<b>max</b>	2547,29	30103,00	17600,00	15106,00	17423,00	17626,00	18658,00	12571,00	6955,00	49,38

En esta tercera tabla, Tabla 15, se puede observar cómo varía el número de puntos por tramo de elevaciones (ver Tabla 8). Los tramos con menos número de puntos son los rangos altos, 5 y 6, elevaciones entre 25 y 35 m. Es decir, existe una cantidad de puntos mayor de la altura sobre el suelo entre 0 y 20m que entre 20 y 35m.

El skewness tiene como dato medio 1,85, es decir un dato positivo. Así pues, se puede decir que la elevación es una curva asimétrica positiva muy marcada. Por otro lado, la curtosis media da valores positivos, así pues, la elevación tiene una distribución leptocúrtica, ya que hay una mayor concentración de los datos en torno a la media (Brown & Robinson, 2002).





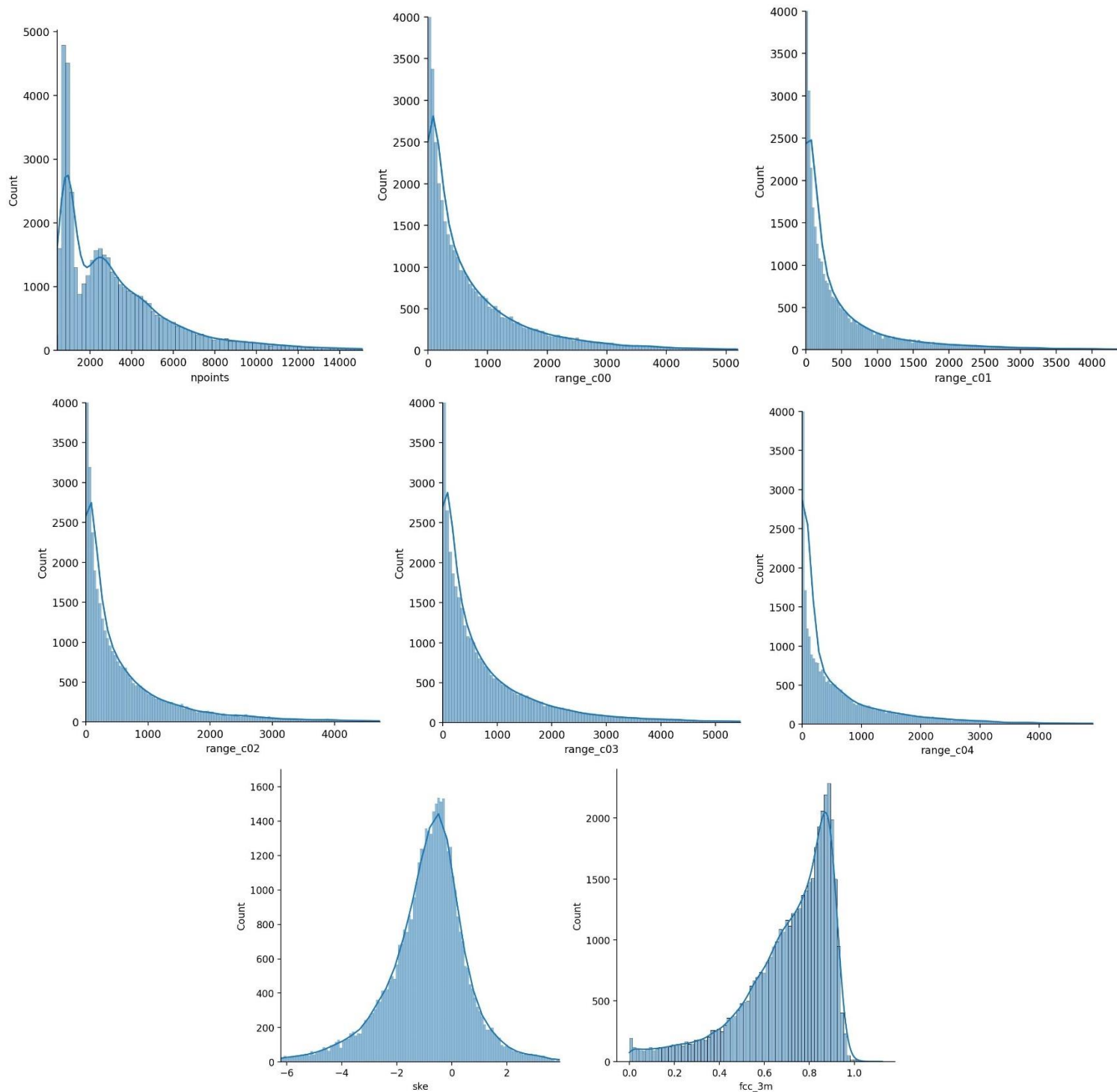


Figura 15: Histogramas de las métricas LiDAR para *F.sylvatica*

Algunas métricas sí que parece que tienen una distribución normal. Como la métrica skewness. Existen otras como, el número de puntos, que no parece que sigan ningún comportamiento normal.

Por otra parte, cada grupo de métricas tiende a componer un histograma parecido, como es de esperar. Las métricas relacionadas con la elevación, por ejemplo, parece que tienen



una distribución normal todas ellas. Para saber estadísticamente si siguen una distribución normal o no habría que realizar un test de normalidad.

Por otro lado a continuación, se representan las estadísticas descriptivas para el *P.sylvestris*:

Tabla 16: Métricas LiDAR relacionadas con la elevación para la especie *P.sylvestris*

	elevation_average	elevation_cv	elevation_highest	elevation_lowest	elevation_p25	
count	15015	15015	15015	15015	15015	
mean	8,47	-	16,58	0,20	5,17	
std	4,83	-	5,81	1,32	5,34	
min	0,00	-	0,00	0,00	0,00	
25%	4,87	0,44	12,56	0,00	0,72	
50%	7,88	0,59	16,23	0,00	3,20	
75%	11,46	0,79	20,51	0,00	8,39	
max	27,45	11,00	53,06	23,52	26,76	
	elevation_p50	elevation_p75	elevation_p90	elevation_p95	elevation_std	fcc_3m
count	15015	15015	15015	15015	15015	15015
mean	9,04	11,69	13,41	14,22	4,41	0,64
std	5,94	5,88	5,84	5,84	1,90	0,21
min	0,00	0,00	0,00	0,00	0,00	0
25%	4,26	7,66	9,44	10,25	3,10	0,53
50%	8,71	11,44	13,14	13,96	4,19	0,69
75%	13,16	15,74	17,36	18,17	5,51	0,79
max	28,67	31,92	43,35	44,33	14,76	1,09

En esta primera tabla, Tabla 16, se puede observar que las elevaciones del pino no superan a las del haya, quedan bastante por debajo, con una elevación media de la métrica elevación media que oscila entre 0 y 27m y con una media de 8,47 (se recuerdan los valores del haya: entre 0 y 48m y media de 12m).

Tal y como ha sucedido con los mínimos de las elevaciones del haya, en esta ocasión también, los valores mínimos están iguales a 0, o muy cercanos a él. Por destacar un valor, se destaca el valor de elevación máxima, 53m.

Hay que señalar que el número de datos para el pino es menor que para el haya. El pino cuenta con un muestreo de 15.015 puntos y el haya con 44.443.

Tabla 17: Métricas LiDAR relacionadas con la intensidad para la especie *P.sylvestris*

	intensity_average	intensity_cv	intensity_highest	intensity_lowest	intensity_std
count	15015	15015	15015	15015	15015
mean	5702,83	-	7421,29	4802,67	436,61
std	465,32	-	1140,55	210,93	222,47
min	4911,00	-	5200,00	4243,00	0,00
25%	5424,50	0,05	6586,00	4691,00	293,00
50%	5600,00	0,07	7204,00	4785,00	384,00
75%	5835,00	0,09	8078,00	4887,00	510,00
max	10101,00	0,32	13421,00	7878,00	2356,00

En las Tabla 17, se observa que las intensidades medias para el pino están entre 5.700 y 10.000. Con valores máximos de intensidad que llegan a 13.000 y mínimos de intensidad que se encuentran entorno a los 200.

Estos valores son muy similares a los del haya.

Tabla 18: Métricas LiDAR relacionadas con la forma para la especie *P.sylvestris*, parte 1

	kur	npoints	range_c00	range_c01	range_c02
<b>count</b>	15015	15015	15015	15015	15015
<b>mean</b>	3,97	4737,28	1418,02	1398,56	1048,21
<b>std</b>	25,58	3290,61	1516,97	1567,03	1420,31
<b>min</b>	0,00	149,00	0,00	0,00	0,00
<b>25%</b>	1,73	2269,50	334,00	266,00	68,00
<b>50%</b>	2,28	4119,00	966,00	872,00	512,00
<b>75%</b>	3,41	6406,00	1999,50	2009,00	1494,00
<b>max</b>	1719,44	36785,00	19928,00	14838,00	22452,00

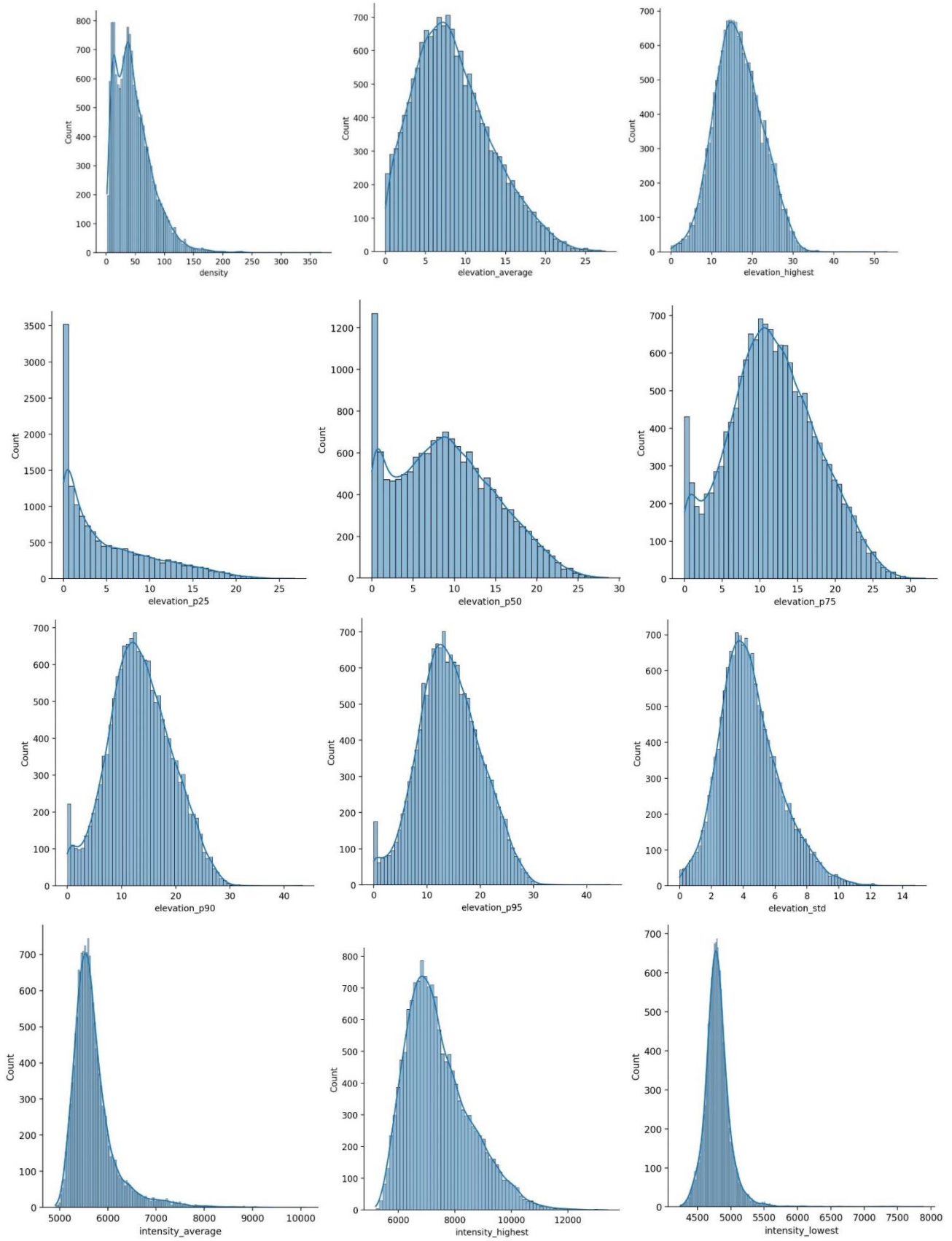
Tabla 19: Métricas LiDAR relacionadas con la forma para la especie *P.sylvestris*, parte 2

	range_c03	range_c04	range_c05	range_c06	ske	density
<b>count</b>	15015	15015	15015	15015	15015	15015
<b>mean</b>	589,08	212,97	29,36	0,61	-0,24	47,37
<b>std</b>	1108,94	668,25	205,24	19,70	1,33	32,91
<b>min</b>	0,00	0,00	0,00	0,00	-5,50	1,49
<b>25%</b>	0,00	0,00	0,00	0,00	-0,91	22,70
<b>50%</b>	34,00	0,00	0,00	0,00	-0,36	41,19
<b>75%</b>	714,00	6,00	0,00	0,00	0,27	64,06
<b>max</b>	12755,00	12555,00	5026,00	1730,00	36,21	367,85

En estas dos últimas tablas, Tabla 18 y Tabla 19, se observan los números de puntos por rango, la densidad, la curtosis y el skewness. La densidad media, 43pts/m<sup>2</sup>, en este caso, es algo superior al haya 33pts/m<sup>2</sup>. Algo a destacar dentro de la densidad es el valor máximo, 367pts/m<sup>2</sup>. Valor que supera también al del haya. De igual manera que ocurre con la especie anterior, la distribución del pino silvestre está limitada a las zonas norte, donde hay un número más alto de pasada y por ende, una densidad mayor.

El skewness en este caso, tiene como dato medio -0,24, es decir un dato negativo. Así pues, se puede decir que la elevación tiene una curva asimétrica negativa. Por otro lado, la curtosis media da valores positivos (3,97), así pues, la elevación en el caso del pino también tiene una distribución leptocúrtica, pero algo menos marcada que el haya (Brown & Robinson, 2002).

Después de comentar los estadísticos se representan sus histogramas:



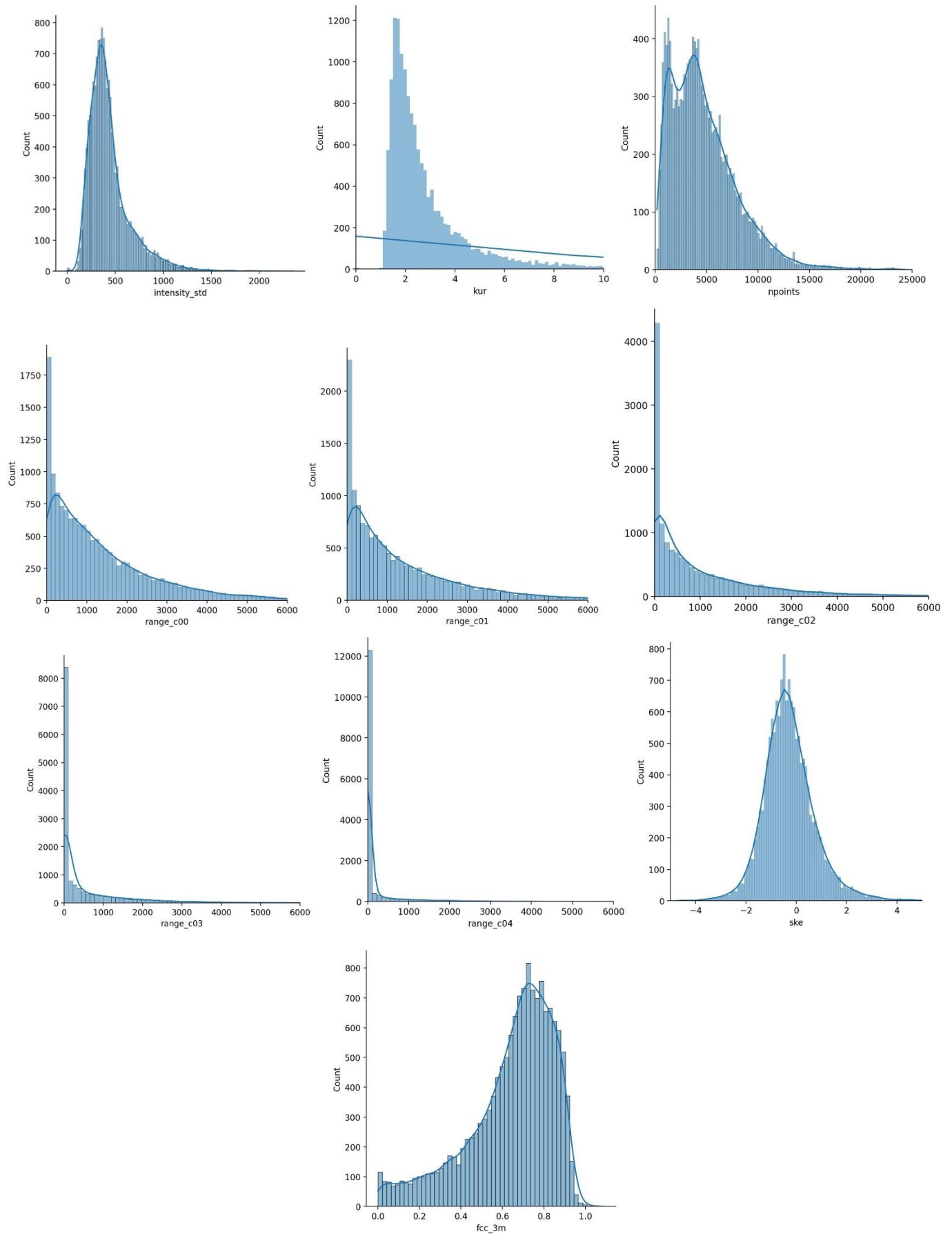


Figura 16: Histogramas de las métricas LiDAR calculadas para *P.sylvestris*

De la misma manera que para el haya, para el pino se han representado los histogramas de las métricas LiDAR. En esta ocasión, las métricas de elevación (excluyendo el percentil 25 y 50), las métricas de intensidad y el skewness presentan histogramas que pueden seguir una distribución normal. En cambio, las métricas relacionadas con el número de puntos no.

#### 4.3. CORRELACIONES:

En este apartado, se muestran los resultados de las correlaciones tanto para el haya como para el pino calculadas para las métricas LiDAR mediante tres métodos diferentes: coeficiente de correlación de Pearson, el de Spearman, y el Tau-b de Kendall. Tal y como se ha visto en el apartado anterior, todas las métricas no siguen una distribución normal. Por ello, se ha visto de gran utilidad calcular mediante los métodos Kendall y Spearman para comparar los resultados.

### 4.3.1. PEARSON

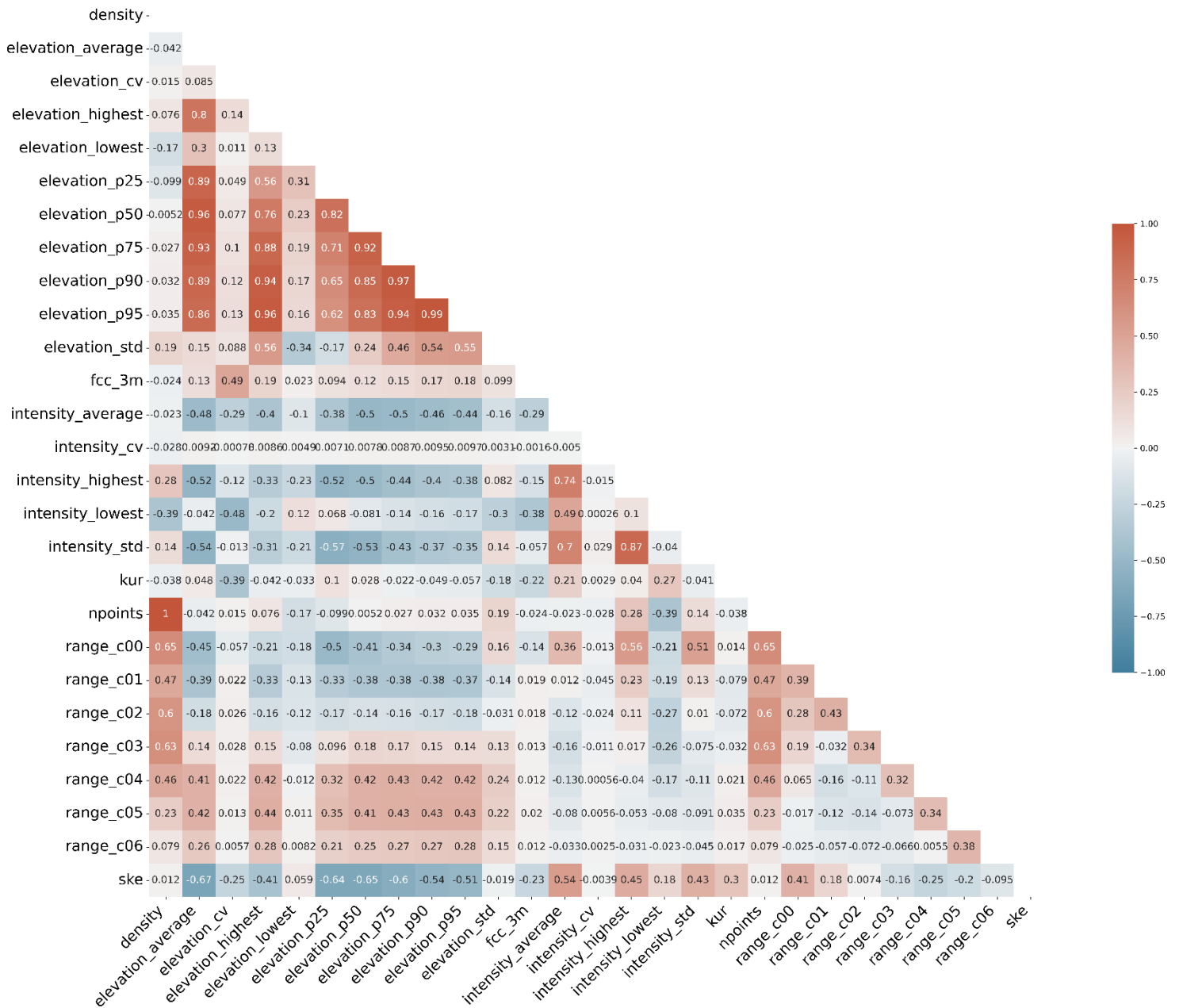


Figura 17: Correlaciones de las métricas LiDAR para *F.sylvatica*, método Pearson.

### 4.3.2. KENDALL

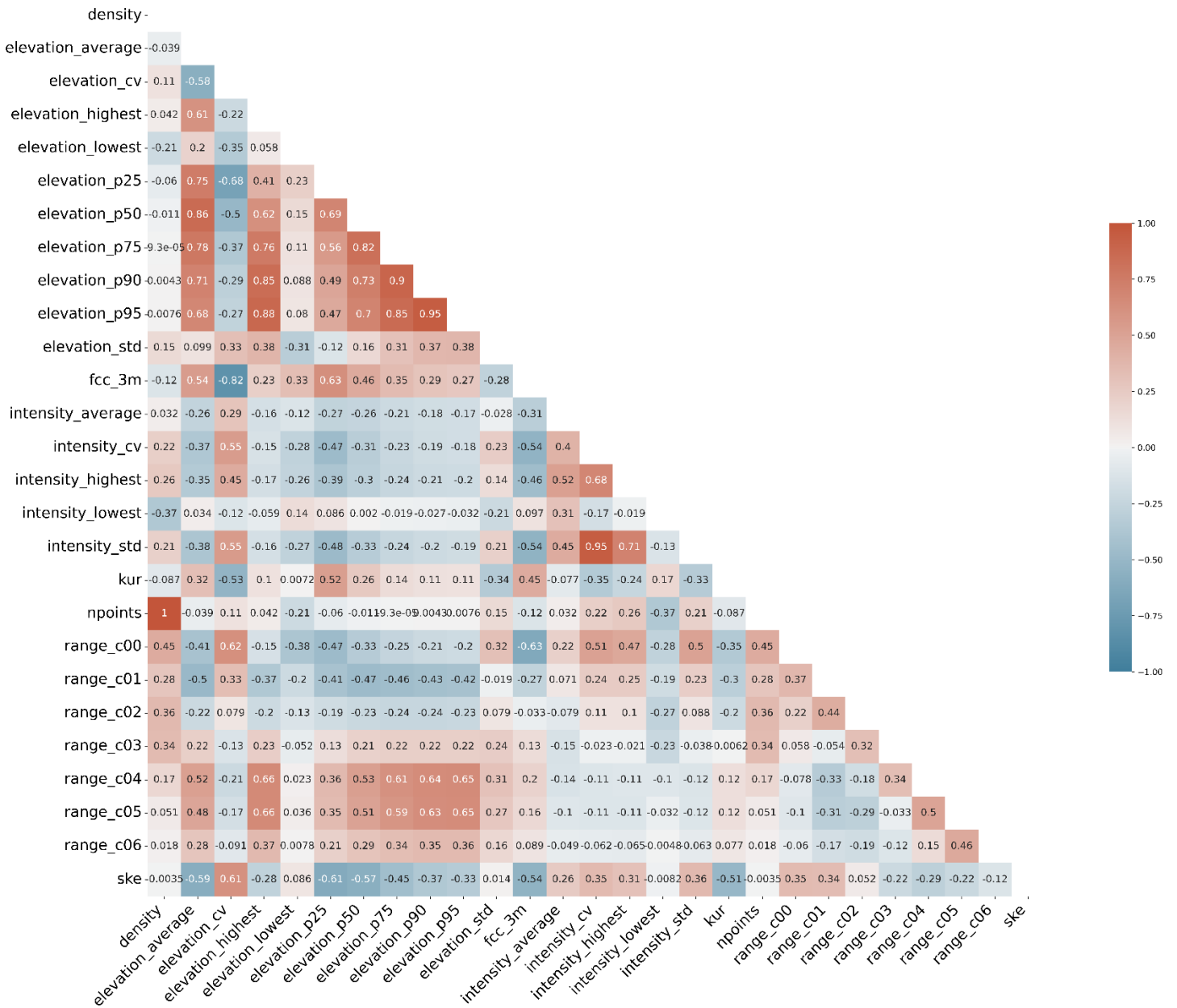


Figura 18: Correlaciones de las métricas LiDAR para *F.sylvatica*, método Kendall

### 4.3.3. SPEARMAN

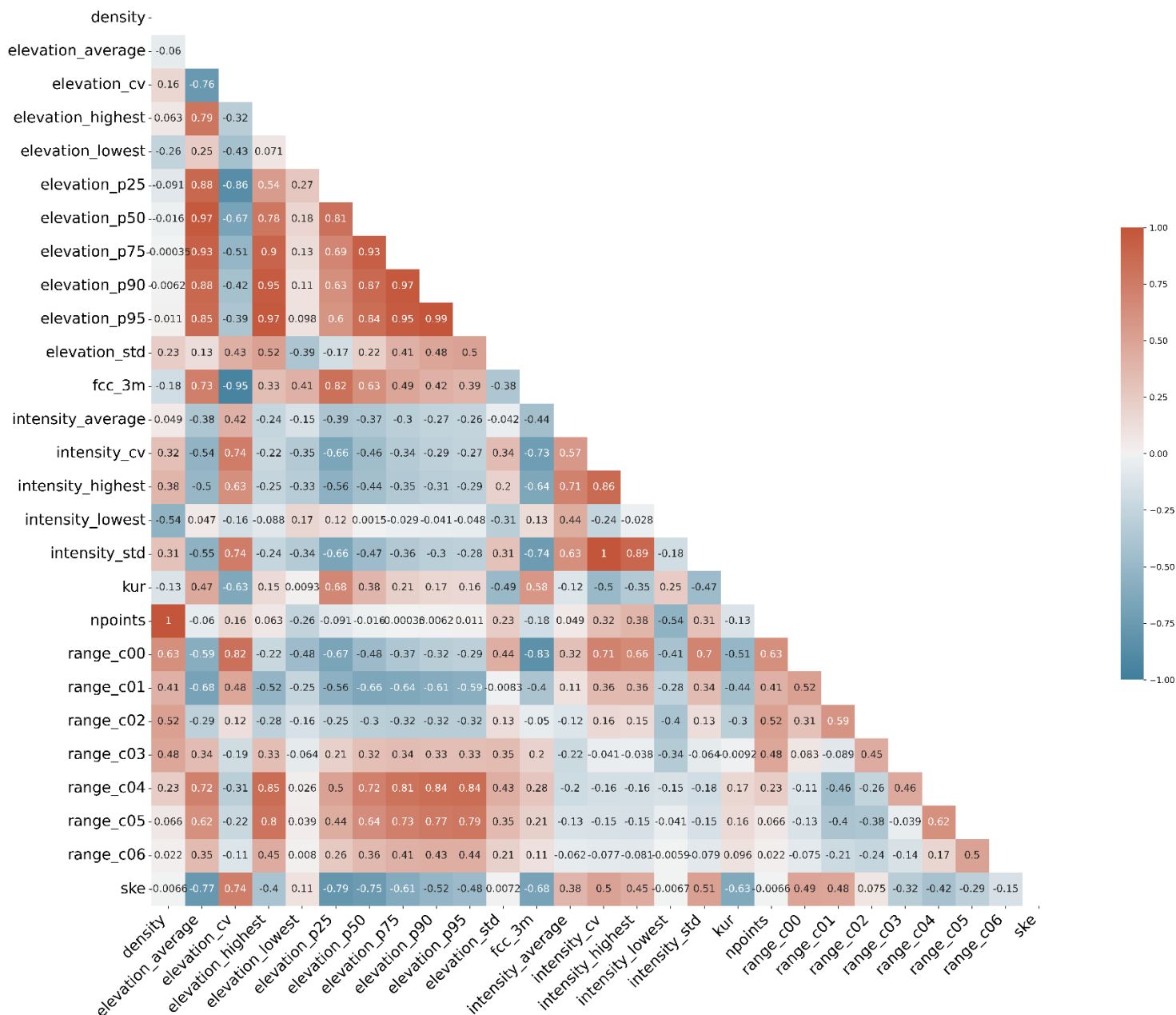


Figura 19: Correlaciones de las métricas LiDAR para F.sylvatica, método Spearman.

En las figuras anteriores, Figura 17, Figura 18 y Figura 19, y en las del ANEXO X; Figura 20 Figura 21 y Figura 22, están representadas las 27 métricas LiDAR.

Para empezar, es visible la similitud de las correlaciones entre los diferentes métodos, aunque cabe destacar algunas diferencias notables en determinadas métricas. Para empezar, existe una correlación más aguda entre la densidad y todas las intensidades en los métodos Kendall y Spearman, cosa que el método Pearson no muestra.

Las principales correlaciones se muestran como era de esperar, entre los percentiles de altura, con una correlación positiva, alrededor de 0,9. También están correlacionados los



rangos de puntos de alturas con los percentiles, con una correlación positiva entre 0,7 y 0,8.

Por otra parte, la fracción de cabida cubierta está altamente correlacionada con las elevaciones, tanto con la media como con los percentiles, no así con la elevación mínima (0,29). También muestra correlaciones negativas con las métricas de intensidad y skewness (-0,81).

Por otra parte, entre FCC y el rango 0 Pearson no muestra ninguna correlación, mientras que con Kendall y Spearman tiene una correlación negativa, valores que oscilan entre 0,6 y 0,8. Todas estas diferencias entre los métodos se deben a la distribución de los datos. Cuando los datos siguen una distribución más normal y las correlaciones son lineales, el método Pearson muestra correlaciones (al igual que los otros), mientras que cuando los datos no siguen una distribución normal y las relaciones son no lineales, entonces, Kendall y Spearman son capaces de mostrar la correlación y Pearson no. Cabe subrayar también que entre los métodos Kendall y Spearman, en general, el método Spearman tiende a mostrar valores más altos de correlación que el método Kendall.

Otros valores a destacar son los valores 1 en algunas de las matrices de correlación. En todas las matrices entre las métricas de nº de puntos y densidad muestran ese valor 1, ya que la densidad es el nº de puntos partido por el área en m<sup>2</sup>. Existe también otro valor 1 entre dos métricas, es entre las métricas de desviación estándar y el coeficiente de variación de la intensidad, pero en este caso, solo se muestra en el haya con el método Spearman, cosa que no sucede en los demás métodos ni con la especie pino, a pesar de que los valores son muy cercanos al 1. Esto es debido a que el coeficiente de variación se calcula a partir de la división entre la desviación estándar con la media.

Como era de esperar, las métricas más relacionadas entre sí son las elevaciones, por una parte, de forma positiva negativa, según la métrica y, por otra parte, los rangos de nº de puntos con las elevaciones, correlaciones más claras con el método Spearman.

Para terminar, la métrica skewness ofrece grandes correlaciones con las elevaciones, excluyendo la elevación mínima y la desviación estándar, y también con la curtosis. Skewness (asimetría) se relaciona con la distribución de puntos en el dosel. Una asimetría positiva, corresponde con una distribución asimétrica hacia la izquierda lo que representa un mayor concentración de puntos a alturas bajas, por debajo de la media, mientras que una asimetría negativa, muestra lo contrario, una mayor concentración de puntos en la parte alta del dosel. La curtosis indica si los puntos se agrupan entorno a la media o no, una distribución leptocúrtica (curtosis positiva) indica una mayor agrupación de datos en torno a la media que en el caso de la distribución normal, en cambio una distribución platicúrtica (curtosis negativa) indica una mayor dispersión de los datos. Esto tiene implicaciones en la estructura de la copa y los rangos de altura en los que se generan los retornos.

#### 4.4. MODELOS DE REGRESION:

Finalmente, en este apartado, se presentan los modelos de regresión calculados para estimar las variables forestales AB, AD y VT para las dos especies forestales.

#### 4.4.1. F. SYLVATICA:

Primeramente se muestran en estas tablas (Tabla 20 Tabla 21 Tabla 22 Tabla 23 Tabla 24 y Tabla 25), los resultados obtenidos para el haya (La casilla de tiempo que aparece en todas las tablas corresponde al tiempo empleado al entrenar el árbol de decisión).

Tabla 20: Resultados de la precisión de los modelos de regresión para el área basimétrica obtenidos con *F.sylvatica*

ÁREA BASIMÉTRICA (m <sup>2</sup> /ha)					
	LIDAR	SENTINEL-1	SENTINEL-2	LIDAR + SENTINEL1	LIDAR + SENTINEL-2
R2train	0,98	0,86	0,95	0,98	0,96
RMSEtrain	1,42	3,97	2,36	1,52	2,07
R2test	0,88	0,82	0,83	0,85	0,74
RMSEtest	3,70	4,43	4,32	4,07	5,39
Tiempo(s)	96,22	11,86	17,50	113,88	717,03

Tabla 21: Mejores parámetros para el árbol de decisión de área basimétrica de *F.sylvatica*

	LIDAR	SENTINEL-1	SENTINEL-2	LIDAR + SENTINEL1	LIDAR + SENTINEL2
<i>max_depth:</i>	None	10	None	None	None
<i>max_features</i>	None	None	None	None	None
<i>min_samples_split</i>	2	10	2	2	2
<i>n_estimators</i>	100	100	100	100	100

En cuanto al área basimétrica, el coeficiente de determinación con mayor precisión lo obtiene el modelo de regresión calculado a partir de las métricas LiDAR, con un  $R^2 = 0,88$ , y el peor, el modelo conjunto de LiDAR con Sentinel-2, con un  $0,74$  (Tabla 20). De todas maneras, todos los modelos parecen estimar con una calidad bastante alta el área basimétrica. En cuanto al error cuadrático medio, el que menor error muestra es el modelo LiDAR, con un error de  $3,7 \text{ m}^2/\text{ha}$ .

Por otra parte, en la mayoría de los modelos de regresión el mejor conjunto de parámetros para entrenar el árbol de decisión ha sido el mismo, a diferencia de Sentinel-1 que en la profundidad máxima y en el número mínimo de observaciones ha sido de 10 (Tabla 21).

Comparando estos datos con la desviación estándar de los datos de las ordenaciones forestales, el área basimétrica tiene una desviación estándar de  $5,26 \text{ m}^2/\text{ha}$ , muy superior al RMSE de LiDAR,  $3,7 \text{ m}^2/\text{ha}$ .

Tabla 22: Resultados de la precisión de los modelos de regresión para la altura dominante obtenidos con *F.sylvatica*

ALTURA DOMINANTE (m)					
	LIDAR	SENTINEL-1	SENTINEL-2	LIDAR + SENTINEL1	LIDAR + SENTINEL-2
R2train	0,96	0,91	0,99	0,99	0,95
RMSEtrain	1,09	1,71	0,66	0,62	1,28
R2test	0,73	0,89	0,90	0,91	0,64
RMSEtest	2,92	1,83	1,75	1,66	3,37
elapsed_time	95,31	11,77	17,73	111,97	635,00

 Tabla 23: Mejores parámetros para el árbol de decisión de altura dominante de *F.sylvatica*

	LIDAR	SENTINEL-1	SENTINEL-2	LIDAR + SENTINEL1	LIDAR + SENTINEL2
<i>max_depth:</i>	None	10	None	None	None
<i>max_features</i>	None	None	None	None	None
<i>min_samples_split</i>	2	10	2	2	2
<i>n_estimators</i>	100	100	100	100	100

En cuanto a la altura dominante, el mejor modelo de regresión parece ser el modelo conjunto de LiDAR con imágenes Sentinel-1, con un  $R^2 = 0,91$  y el peor, el conjunto de LiDAR con imágenes Sentinel-2 (Tabla 22). Como RMSE obtiene un 1,66 m de error. En esta variable cabe destacar que los dos modelos de conjuntos de datos muestran valores diferentes que cuando se calculan por separado. Además, el modelo LiDAR y el modelo Sentinel-1 tienen un  $R^2$  más bajo que en el modelo con su combinación. Cosa que no ocurre con Sentinel-2 y LiDAR.

Por otra parte, los mejores parámetros para entrenar el árbol de decisión vuelven a ser los mismos que en la anterior variable (Tabla 23). Hay una similitud entre todos excepto con Sentinel-1, que sigue siendo mejor 10 el número mínimo de observaciones y la máxima profundidad.

Comparando estos datos con la desviación estándar de los datos de las ordenaciones forestales, la altura dominante tiene una desviación estándar de 10,3 m, muy superior al RMSE de LiDAR, 2,92m.

Tabla 24: Resultados de la precisión de los modelos de regresión para volumen total obtenidos con *F.sylvatica*

VOLUMEN TOTAL					
	LIDAR	SENTINEL-1	SENTINEL-2	LIDAR + SENTINEL1	LIDAR + SENTINEL-2
R2train	0,91	0,66	0,56	0,90	0,92
RMSEtrain	2997,78	5931,55	6802,02	3205,49	2866,26
R2test	0,51	0,19	0,24	0,44	0,48
RMSEtest	8939,70	11478,37	11147,87	9595,95	9237,51
Tiempo(s)	107,22	6,69	11,92	121,23	784,82

Tabla 25: Mejores parámetros para el árbol de decisión de volumen de *F.sylvatica*

	LIDAR	SENTINEL-1	SENTINEL-2	LIDAR + SENTINEL1	LIDAR + SENTINEL2
<i>max_depth:</i>	None	10	10	None	None
<i>max_features</i>	None	Sqrt	None	none	none
<i>min_samples_split</i>	2	10	20	2	2
<i>n_estimators</i>	100	100	100	100	100

Finalmente, las estimaciones para la variable volumen son claramente peores que las dos anteriores. El  $R^2$  óptimo en este caso no llega al mínimo de los modelos de regresión de las otras dos variables, teniendo el LiDAR el mejor  $R^2$ , con un 0,51. Por otra parte, destacar que los modelos de regresión combinados mejoran los resultados de los modelos de regresión de Sentinel-1 y Sentinel-2 con una clara diferencia.

El RMSE muestra unos errores elevados de volumen, desde 9.000m<sup>3</sup>/ha hasta 11.000m<sup>3</sup>/ha, que es en general superior a la desviación estándar de las ordenaciones, 8.635m<sup>3</sup>/ha.

#### 4.4.2. P. SYLVESTRIS

En este apartado se muestran los resultados obtenidos para el pino en las siguientes tablas (Tabla 26 Tabla 27 Tabla 28 Tabla 29 Tabla 30 Tabla 31):

Tabla 26: Resultados de la precisión de los modelos de regresión para el área basimétrica obtenidos con *P. sylvestris*.

ÁREA BASIMÉTRICA					
	LIDAR	SENTINEL-1	SENTINEL-2	LIDAR + SENTINEL1	LIDAR + SENTINEL-2
R2train	0,99	0,94	0,99	0,99	0,96
RMSEtrain	1,56	3,78	1,67	1,55	3,18
R2test	0,93	0,91	0,92	0,93	0,69
RMSEtest	4,09	4,52	4,34	4,07	8,36

<b>elapsed_time</b>	27,27	3,84	5,76	29,93	10,84
---------------------	-------	------	------	-------	-------

 Tabla 27: Mejores parámetros para el árbol de decisión de área basimétrica de *P.sylvestris*

	<b>LIDAR</b>	<b>SENTINEL-1</b>	<b>SENTINEL-2</b>	<b><u>LIDAR + SENTINEL1</u></b>	<b><u>LIDAR + SENTINEL2</u></b>
<i>max_depth:</i>	None	10	None	None	None
<i>max_features</i>	None	None	None	none	none
<i>min_samples_split</i>	2	2	2	2	2
<i>n_estimators</i>	100	100	100	100	100

El mejor modelo de regresión para estimar el área basimétrica del pino parece ser el LiDAR, y también el modelo combinado de LiDAR y Sentinel-1 con un  $R^2$  de 0,93 para los dos y un RMSE de 4m<sup>2</sup>/ha.

 Tabla 28: Resultados de la precisión de los modelos de regresión para la altura dominante obtenidos con *P. sylvestris*.

<b>ALTURA DOMINANTE</b>					
	<b>LIDAR</b>	<b>SENTINEL-1</b>	<b>SENTINEL-2</b>	<b>LIDAR + SENTINEL1</b>	<b>LIDAR + SENTINEL-2</b>
<b>R2train</b>	0,96	0,88	0,97	0,96	0,73
<b>RMSEtrain</b>	1,22	2,15	1,19	1,21	3,26
<b>R2test</b>	0,91	0,89	0,88	0,91	0,65
<b>RMSEtest</b>	1,56	1,73	1,8	1,55	3,10
<b>elapsed_time</b>	5,46	1,04	0,69	5,52	1,58

 Tabla 29: Mejores parámetros para el árbol de decisión de altura dominante de *P.sylvestris*

	<b>LIDAR</b>	<b>SENTINEL-1</b>	<b>SENTINEL-2</b>	<b><u>LIDAR + SENTINEL1</u></b>	<b><u>LIDAR + SENTINEL2</u></b>
<i>max_depth:</i>	None	10	None	None	None
<i>max_features</i>	Sqrt	Log2	Sqrt	Log2	Sqrt
<i>min_samples_split</i>	2	10	20	2	20
<i>n_estimators</i>	100	50	100	100	100

Para la altura dominante del pino, el mejor modelo de regresión es LiDAR y el modelo de regresión combinado de LiDAR con Sentinel-1, al igual que en el modelo de área basimétrica.

Tabla 30: Resultados de la precisión de los modelos de regresión para el volumen obtenidos con *P. sylvestris*

VOLUMEN TOTAL					
	LIDAR	SENTINEL-1	SENTINEL-2	LIDAR + SENTINEL1	LIDAR + SENTINEL-2
R2train	0,96	0,87	0,91	0,96	0,64
RMSEtrain	42,17	78,10	63,88	41,39	128,26
R2test	0,77	0,71	0,67	0,76	0,55
RMSEtest	108,12	120,00	129,74	109,64	150,14
Tiempo (s)	27,38	1,87	5,10	30,57	1,84

Tabla 31: Mejores parámetros para el árbol de decisión de volumen de *P.sylvestris*

	LIDAR	SENTINEL-1	SENTINEL-2	LIDAR + SENTINEL1	LIDAR + SENTINEL2
<i>max_depth:</i>	None	10	5	None	5
<i>max_features</i>	None	None	None	none	None
<i>min_samples_split</i>	2	2	20	2	20
<i>n_estimators</i>	100	50	50	100	50

En cuanto al volumen estimado para el pino, destacan los modelos de regresión de LiDAR y el combinado de LiDAR más Sentinel-1, con un  $R^2$  de 0,77 y 0,76 respectivamente. Generalmente, los valores son más bajos que en el resto de las variables, pero no tan bajos como lo son en el haya.

## 5. CONCLUSIONES

El objetivo principal de esta investigación fue explorar la viabilidad de utilizar datos LiDAR para estimar tres variables forestales, área basimétrica, altura dominante y volumen total, en masas de haya y pino silvestre, mediante el ajuste de modelos de regresión Random Forests. Los modelos obtenidos han sido además comparados con otros basados en otras fuentes de información, Sentinel-1 y Sentinel-2.

En esta investigación se han calculado 27 métricas LiDAR diferentes, que muestran entre sí diferentes grados de correlación (lineal en unos casos y no lineal en otros) y cierta complementariedad. Los modelos de regresión Random Forests construidos con estas métricas han obtenido resultados satisfactorios para las variables área basimétrica y altura dominante, principalmente para el pino silvestre (AB:  $R^2=0,93$  RMSE=4,09m<sup>2</sup>/ha; AD:  $R^2=0,91$  RMSE=1,56m). Para el haya los resultados de estas dos variables han sido positivos, pero no tan precisos (AB:  $R^2=0,88$  RMSE=3,70m; AD:  $R^2=0,73$  RMSE=2,92m<sup>2</sup>/ha). El volumen total, en cambio ha proporcionado resultados más pobres tanto para el haya (VT:  $R^2=0,51$  RMSE=8939,70m<sup>3</sup>/ha) como para el pino (VT  $R^2=0,77$  RMSE=108,77m<sup>3</sup>/ha). El ajuste de modelos basados en datos Sentinel-1 y Sentinel-2 ha proporcionado, en general unos resultados algo peores que el de los modelos basados en LiDAR. También hay que señalar que los modelos combinados de LiDAR+Sentinel-1 y LiDAR+Sentinel-2 proporcionan en general rendimientos similares a los basados únicamente en datos LiDAR, por lo que no parece interesante incorporar al análisis esta información adicional.

Es necesario reconocer que la metodología aquí aplicada tiene margen de mejora y se podrían explorar otras alternativas como el realizar un análisis de importancia de variables en RF para descartar aquellas que muestren ser irrelevantes, también se pueden explorar otras métricas LiDAR distintas, se podría trabajar con las series completas de Sentinel-1 y Sentinel-2 en lugar de con los compuestos anuales, y también se podrían probar otras herramientas y algoritmos de regresión.

Con todo, los resultados de este estudio muestran un camino ilusionante para integrar estas herramientas en la gestión forestal, contar con datos LiDAR, cada vez más precisos e imágenes multiespectrales y radar gratuitas constituyen una alternativa para mejorar el conocimiento de las masas forestales y las técnicas tradicionales de inventario. Estas herramientas basadas en observaciones remotas podrán ser en un futuro cercano metodologías eficaces y eficientes plenamente operativas en el sector forestal.

## 6. BIBLIOGRAFÍA

- American Society for Photogrammetry and Remote Sensing, ASPRS. (2019). LAS Specification 1.4 - R14. [http://www.asprs.org/wp-content/uploads/2019/03/LAS\\_1\\_4\\_r14.pdf](http://www.asprs.org/wp-content/uploads/2019/03/LAS_1_4_r14.pdf)
- Ashraf, I. M., Zhao, Z., Bourque, P.-A. C., MacLean, D. A., & Meng, F.-R. (2013). Integrating biophysical controls in forest growth and yield predictions with artificial intelligence technology. *Canadian Journal of Forest Research*, 43(12), 1162–1171. <https://doi.org/10.1139/cjfr-2013-0090>
- Assman, E. (1970). *The Principles of Forest Yield Study*. Pergamon Press. <https://doi.org/10.1016/B978-0-08-006658-5.50002-9>
- Belgiu, M., & Drăgu, L. (2016). Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*, 114, 24–31. <https://doi.org/10.1016/j.isprsjprs.2016.01.011>
- Bergen, K. M., & Dronova, I. (2007). Observing succession on aspen-dominated landscapes using a remote sensing-ecosystem approach. *Landscape Ecology*, 22(9), 1395–1410. <https://doi.org/10.1007/s10980-007-9119-1>
- Breiman, L. (2001) Random Forests. *Machine Learning* 45, 5–32. <https://doi.org/10.1023/A:1010933404324>.
- Breiman, L. (2001). Using iterated bagging to debias regressions. *Machine Learning*, 45(3), 261–277. <https://doi.org/10.1023/A:1017934522171>
- Brown, C. A., & Robinson, D. M. (2002). Skewness and kurtosis implied by option prices: A correction. *Journal of Financial Research*, 25(2), 279–282. <https://doi.org/10.1111/1475-6803.t01-1-00008>
- Calders, K., Schenkels, T., Bartholomeus, H., Armston, J., Verbesselt, J., & Herold, M. (2015). Monitoring spring phenology with high temporal resolution terrestrial LiDAR measurements. *Agricultural and Forest Meteorology*, 203, 158–168. <https://doi.org/10.1016/j.agrformet.2015.01.009>
- Castillo-Santiago, M. A., Ricker, M., & de Jong, B. H. J. (2010). Estimation of tropical forest structure from spot-5 satellite images. *International Journal of Remote Sensing*, 31(10), 2767–2782. <https://doi.org/10.1080/01431160903095460>
- Coomes, D. A., Dalponte, M., Jucker, T., Asner, G. P., Banin, L. F., Burslem, D. F. R. P., Lewis, S. L., Nilus, R., Phillips, O. L., Phua, M.-H., Phua, M.-H., & Qie, L. (2017). Area-based vs tree-centric approaches to mapping forest carbon in Southeast Asian forests from airborne laser scanning data. *Remote Sensing of Environment*, 194, 77–88. <https://doi.org/10.1016/j.rse.2017.03.017>
- Chuevico, E. (1995) *Fundamentos de la teledetección espacial (segunda edición)*. Edifiones RIALP S A.
- Dubayah, R. O., Sheldon, S. L., Clark, D. B., Hofton, M. A., Blair, J. B., Hurtt, G. C., & Chazdon, R. L. (2010). Estimation of tropical forest height and biomass dynamics using lidar remote sensing at la Selva, Costa Rica. *Journal of Geophysical Research: Biogeosciences*, 115(2). <https://doi.org/10.1029/2009JG000933>



- Evans, J. S., Hudak, A. T., Faux, R., & Smith, A. M. S. (2009). Discrete return lidar in natural resources: Recommendations for project planning, data processing, and deliverables. *Remote Sensing*, 1(4), 776–794. <https://doi.org/10.3390/rs1040776>
- Falkowski, M. J., Evans, J. S., Martinuzzi, S., Gessler, P. E., & Hudak, A. T. (2009). Characterizing forest succession with lidar data: An evaluation for the Inland Northwest, USA. *Remote Sensing of Environment*, 113(5), 946–956. <https://doi.org/10.1016/j.rse.2009.01.003>
- Fassnacht, F. E., Poblete-Olivares, J., Rivero, L., Lopatin, J., Ceballos-Comisso, A., & Galleguillos, M. (2021). Using Sentinel-2 and canopy height models to derive a landscape-level biomass map covering multiple vegetation types. *International Journal of Applied Earth Observation and Geoinformation*, 94. <https://doi.org/10.1016/j.jag.2020.102236>
- Gigović, L., Pourghasemi, H. R., Drobnyak, S., & Bai, S. (2019). Testing a new ensemble model based on SVM and random forest in forest fire susceptibility assessment and its mapping in Serbia's Tara National Park. *Forests*, 10(5). <https://doi.org/10.3390/f10050408>
- Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., & Moore, R. (2017). Google Earth Engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*, 202, 18–27. <https://doi.org/10.1016/j.rse.2017.06.031>
- Hall, F. G., Bergen, K., Blair, J. B., Dubayah, R., Houghton, R., Hurtt, G., Kelldorfer, J., Lefsky, M., Ranson, J., Saatchi, S., Shugart, H. H., & Wickland, D. (2011). Characterizing 3D vegetation structure from space: Mission requirements. *Remote Sensing of Environment*, 115(11), 2753–2775. <https://doi.org/10.1016/J.RSE.2011.01.024>
- Houghton, R. A., Hall, F., & Goetz, S. J. (2009). Importance of biomass in the global carbon cycle. *Journal of Geophysical Research: Biogeosciences*, 114(3). <https://doi.org/10.1029/2009JG000935>
- Hu, T., Sun, Y., Jia, W., Li, D., Zou, M., & Zhang, M. (2021). Study on the estimation of forest volume based on multi-source data. *Sensors*, 21(23). <https://doi.org/10.3390/s21237796>
- Huang, W., Min, W., Ding, J., Liu, Y., Hu, Y., Ni, W., & Shen, H. (2022). Forest height mapping using inventory and multi-source satellite data over Hunan Province in southern China. *Forest Ecosystems*, 9. <https://doi.org/10.1016/j.fecs.2022.100006>
- Hudak, A. T., Evans, J. S., & Smith, A. M. S. (2009). LiDAR utility for natural resource managers. *Remote Sensing*, 1(4), 934–951. <https://doi.org/10.3390/rs1040934>
- Lefsky, M. A., Cohen, W. B., Harding, D. J., Parker, G. G., Acker, S. A., & Gower, S. T. (2002). Lidar remote sensing of above-ground biomass in three biomes. *Global Ecology and Biogeography*, 11(5), 393–399. <https://doi.org/10.1046/j.1466-822x.2002.00303.x>
- Li, Y., Li, C., Li, M., & Liu, Z. (2019). Influence of variable selection and forest type on forest aboveground biomass estimation using machine learning algorithms. *Forests*, 10(12). <https://doi.org/10.3390/F10121073>
- Lim, K., Treitz, P., Wulder, M., St-Onge, B., & Flood, M. (2003). LiDAR remote sensing of forest structure. *Progress in Physical Geography*, 27(1), 88–106. <https://doi.org/10.1191/0309133303pp360ra>

- McElhinny, C., Gibbons, P., Brack, C., & Bauhus, J. (2005). Forest and woodland stand structural complexity: Its definition and measurement. *Forest Ecology and Management*, 218(1–3), 1–24. <https://doi.org/10.1016/j.foreco.2005.08.034>
- Næsset, E., & Økland, T. (2002). Estimating tree height and tree crown properties using airborne scanning laser in a boreal nature reserve. *Remote Sensing of Environment*, 79(1), 105–115. [https://doi.org/10.1016/S0034-4257\(01\)00243-7](https://doi.org/10.1016/S0034-4257(01)00243-7)
- Pal, M. (2005). Random forest classifier for remote sensing classification. *International Journal of Remote Sensing*, 26(1), 217–222. <https://doi.org/10.1080/01431160412331269698>
- Pang, Y., Li, Z., Meng, S., Jia, W., & Liu, L. (2016). Subtropical forest biomass estimation using airborne LiDAR and hyperspectral data. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 41, 747–749. <https://doi.org/10.5194/isprsarchives-XLI-B8-747-2016>
- Phua, M.-H., Hue, S. W., Ioki, K., Hashim, M., Bidin, K., Musta, B., Suleiman, M., Yap, S. W., & Maycock, C. R. (2016). Estimating logged-over lowland rainforest aboveground biomass in sabah, Malaysia using airborne LiDAR data. *Terrestrial, Atmospheric and Oceanic Sciences*, 27(4), 481–489. [https://doi.org/10.3319/TAO.2016.01.06.02\(ISRS\)](https://doi.org/10.3319/TAO.2016.01.06.02(ISRS))
- Puliti, S., Breidenbach, J., Schumacher, J., Hauglin, M., Klingenberg, T. F., & Astrup, R. (2021). Above-ground biomass change estimation using national forest inventory data with Sentinel-2 and Landsat. *Remote Sensing of Environment*, 265. <https://doi.org/10.1016/j.rse.2021.112644>
- Roberts, J. W., Tesfamichael, S., Gebreslasie, M., van Aardt, J., & Ahmed, F. B. (2007). Forest structural assessment using remote sensing technologies: An overview of the current state of the art. *Southern Hemisphere Forestry Journal*, 69(3), 183–203. <https://doi.org/10.2989/SHFJ.2007.69.3.8.358>
- Saxena, S. (2020). A Beginner's Guide to Random Forest Hyperparameter Tuning. <https://www.analyticsvidhya.com/blog/2020/03/beginners-guide-random-forest-hyperparameter-tuning/>
- Serrada (1995) Avance Apuntes de Selvicultura. E.U.I.T.Forestal. Madrid.
- Shugart, H. H., Saatchi, S., & Hall, F. G. (2010). Importance of structure and its measurement in quantifying function of forest ecosystems. *Journal of Geophysical Research: Biogeosciences*, 115(4). <https://doi.org/10.1029/2009JG000993>
- Silva, A. G. P., Görgens, E. B., Campoe, O. C., Alvares, C. A., Stape, J. L., & Rodriguez, L. C. E. (2015). Assessing biomass based on canopy height profiles using airborne laser scanning data in eucalypt plantations. *Scientia Agricola*, 72(6), 504–512. <https://doi.org/10.1590/0103-9016-2015-0070>
- Torre-Tojal, L., Bastarrika, A., Boyano, A., Lopez-Guede, J. M., & Graña, M. (2022). Above-ground biomass estimation from LiDAR data using random forest algorithms. *Journal of Computational Science*, 58. <https://doi.org/10.1016/j.jocs.2021.101517>
- Wallace, L., Lucieer, A., Watson, C., & Turner, D. (2012). Development of a UAV-LiDAR system with application to forest inventory. *Remote Sensing*, 4(6), 1519–1543. <https://doi.org/10.3390/rs4061519>

- Wernicke, J., Seltmann, C. T., Wenzel, R., Becker, C., & Körner, M. (2022). Forest canopy stratification based on fused, imbalanced and collinear LiDAR and Sentinel-2 metrics. *Remote Sensing of Environment*, 279. <https://doi.org/10.1016/j.rse.2022.113134>
- Wulder, M. A., & Coops, N. C. (2014). Satellites: Make Earth observations open access. *Nature*, 513(7516), 30–31. <https://doi.org/10.1038/513030a>
- Yu, X., Hyypä, J., Holopainen, M., & Vastaranta, M. (2010). Comparison of area-based and individual tree-based methods for predicting plot-level forest attributes. *Remote Sensing*, 2(6), 1481–1495. <https://doi.org/10.3390/rs2061481>
- Zimble, D. A., Evans, D. L., Carlson, G. C., Parker, R. C., Grado, S. C., & Gerard, P. D. (2003). Characterizing vertical forest structure using small-footprint airborne LiDAR. *Remote Sensing of Environment*, 87(2–3), 171–182. [https://doi.org/10.1016/S0034-4257\(03\)00139-1](https://doi.org/10.1016/S0034-4257(03)00139-1)
- Zolkos, S. G., Goetz, S. J., & Dubayah, R. (2013). A meta-analysis of terrestrial aboveground biomass estimation using lidar remote sensing. *Remote Sensing of Environment*, 128, 289–298. <https://doi.org/10.1016/j.rse.2012.10.017>

## 7. ANEXOS

### I. ANEXO I:

→ [Composición de Mosaicos LiDAR:](#)

```
import os
from osgeo import gdal
import glob
import subprocess
import numpy
import matplotlib.pyplot as plt
import subprocess
from pathlib import Path
from datetime import datetime
from time import time

#####
#####
## LISTA DE METRICAS

directorio =
r'Y:\LiDAR\2017_metricas2\536\536_4713_2017_NAV_EPSG25830'
ruta2 = os.listdir(directorio)

TIFS = [s for s in ruta2 if ".tfw" not in s and ".tif.aux" not in s
and ".txt" not in s]

matricas = [izenak[izenak.rfind("0")+2 :izenak.rfind(".")] for izenak
in TIFS]

#####
#####
## FUNCION DEPURAR

density_depura = []

def depurar(lista): ##Definir una funcion para depurar los archivos
que no estén vacíos.
    lista_depura = []
    for x in lista:
        file_size = Path(x).stat().st_size
        if file_size >0: lista_depura.append(x)
    return lista_depura ##tenemos que poner return porque sino no nos
devuelve la lista que queremos crear.

#####
#####
## MERGED DE LAS METRICAS
#####
#####
```

```
directorio2 = r'Y:\LiDAR\2017_metricas2'
carpetas = glob.glob(os.path.join(directorio2, "*", "*"))

lista = glob.glob(os.path.join(directorio2, "*", "*", "*.tif"))
for metrica in matricas:
    start_time = time()
    print(metrica)
    tifs= []
    tiemp = []
    for aldagai in lista:
        if metrica in aldagai:
            tifs.append(aldagai)
    g = gdal.Warp(r'Y:\\LiDAR\\Merge_v2\\' + metrica + '.tif', tifs,
format="GTiff")
    print(metrica + " acabada")
    elapsed_time = time() - start_time
    tiempo = elapsed_time/60
    tiemp.append(tiempo)
```

## II. ANEXO II:

→ [Filtrado de los datos de campo:](#)

```
import os
import subprocess
import glob
import pandas as pd
import geopandas as gpd
import pyproj
import numpy as np
import shapely.wkt

directorio = r'C:\\XABIER\\TFM\\01-DATOS'
data = gpd.read_file('C:\\XABIER\\TFM\\01-
DATOS\\FORESTOBS\\f_masa.shp')
data_especie = pd.read_excel(r'C:\\XABIER\\TFM\\01-
DATOS\\FORESTOBS\\t_aprovechamientos.xlsx')
nombre = pd.read_excel(r'C:\\XABIER\\TFM\\01-
DATOS\\FORESTOBS\\t_especie.xlsx')
rodal = gpd.read_file('C:\\XABIER\\TFM\\01-
DATOS\\FORESTOBS\\f_rodal.shp')

muestra_rodal = rodal.iloc[0:6, :]
muestra_rodal.to_excel(os.path.join(directorio, 'muestra_datos_iniciales_
s_frodal.xlsx'), index = True)
muestra_data = data.iloc[0:6, :]
muestra_data.to_excel(os.path.join(directorio, 'muestra_datos_iniciales_
_fmasa.xlsx'), index = True)

data.columns
##1.- PONER LOS ZEROS A NULL Y ELIMINAR
data['Area_Basim'].eq(0).any() ##Comprobamos que hay ceros
data['Altura_Dom'].eq(0).any()
data['Volumen_To'].eq(0).any()
```

```
#Cambiamos los ceros a null
data.loc[data['Area_Basim'] == 0, 'Area_Basim']= np.nan
data.loc[data['Altura_Dom'] == 0, 'Altura_Dom']= np.nan
data.loc[data['Volumen_To'] == 0, 'Volumen_To']= np.nan

##pasar a pandas dataframe --> data1 = pd.DataFrame(data)

##Comprobamos que se han convertido en null.
data['Area_Basim'].value_counts(ascending=True)
data['Area_Basim'].eq(0).any()
data['Altura_Dom'].eq(0).any()
data['Volumen_To'].eq(0).any()

#De forma global
data.eq(0).any()

##Eliminar las columnas que contienen nulls.
masas = data[['id','IDPOM','SP1',
'Area_Basim','Altura_Dom','Volumen_To','geometry']]

datuak = masas.dropna()
#Se quedan 11.124 filas de un total de 31.088

##Extraigo solo los poligonos para hacer el muestreo

poligonos = datuak[['id','SP1','geometry']]
poligonos2 = poligonos.reset_index()
poligonos3 = poligonos2.drop(['index'], axis=1)
poligonos2['FID'] = np.arange(len(poligonos2))
directorio = r'C:\\XABIER\\TFM\\01-DATOS'
poligonos2.to_csv(os.path.join(directorio,'poligonos_4.csv'), index =
True)
datuak.to_file(os.path.join(directorio,'poligonostrabajo2.shp'),driver
='ESRI Shapefile')

##2.- CALCULAR AREA
datuak = datuak.to_crs("EPSG:25830")
datuak['area']= datuak['geometry'].area/10**6 ##creamos nueva columna
con el area de cada poligono en km2

##3.- CAMBIAMOS EL CODIGO DE ESPECIE POR SU NOMBRE
especie_nombre = nombre[['nombre_comun','codigo']]
numeros = especie_nombre.rename(columns={'codigo':'SP1'})

especie_numeros_nombre = pd.merge(numeros, datuak, on="SP1")

ejemplo1 =
especie_numeros_nombre[['nombre_comun','Area_Basim','Altura_Dom','Volu
men_To']]
prueba = ejemplo1.groupby("nombre_comun", as_index = True).describe()

directorio = r'C:\\XABIER\\TFM\\01-DATOS'
prueba.to_excel(os.path.join(directorio,'prueba1.xlsx'), index = True)
```

```
##Ahora coinciden el número de muestras por especie

## 4.- CÁLCULO DEL AREA POR ESPECIE
solo_area = datuak[['SP1','area']]
area_especie_nombre = pd.merge(numeros, solo_area, on="SP1")

area_juntado = area_especie_nombre.groupby("nombre_comun", as_index =
True).sum()

## 5.- SUMAR LA COLUMNA ÁREA AL DATAFRAME GENERAL
### 5.1. Cambiar el índice a una columna

prueba['nombre'] = prueba.index
area_juntado['nombre'] = area_juntado.index
area_unir = area_juntado[['area','nombre']]

### 5.1. Unir

tablas_datos_finales = pd.merge(area_unir, prueba, on="nombre")

## 6. ENVIAR A EXCEL

directorio = r'C:\\XABIER\\TFM\\01-DATOS'
tablas_datos_finales.to_excel(os.path.join(directorio, 'DatosFINALES.xlsx'), index = True)
```

### III. ANEXO III:

#### → [Extracción métricas LiDAR](#)

```
from datetime import datetime
import geopandas as gpd
import rasterio
import pandas as pd
import glob
import os

#DEFINICIÓN DE CONSTANTES:
directorio =
r'C:\Users\goldaraz.112790\Desktop\muestrero\muestrero_hayas'
dirfin = r'C:\Users\goldaraz.112790\Desktop\Prueba_python\TFM\Datos'

####
#1.-LECTURA DE DATOS:
print('Preparando datos...')

pointData5 =
gpd.read_file(os.path.join(directorio, 'DataHayaMuestreo_05_bien.shp'))
x5, y5 = [point.xy[0][0] for point in pointData5['geometry']],
[point.xy[1][0] for point in pointData5['geometry']] ##Sacar las
coordenadas x e y de cada punto del muestreo

Volumen_To = list(pointData5['Volumen_To'])
Altura_Dom = list(pointData5['Altura_Dom'])
Area_Basim = list(pointData5['Area_Basim'])
```

```
qs_Volumen = list(pointData5['qs_Volumen'])
qs_Altura = list(pointData5['qs_Altura'])
qs_Area = list(pointData5['qs_Area'])

infoini5 = {'x': x5, 'y': y5, 'Volumen_To': Volumen_To, 'Altura_Dom':
Altura_Dom, 'Area_Basim': Area_Basim, 'qs_Volumen': qs_Volumen,
'qs_Altura': qs_Altura, 'qs_Area': qs_Area } ##definicion diccionario
de 8 columnas. con las coordenadas(x e y) y las variables forestales.

df5 = pd.DataFrame(data=infoini5) ##generar dataframe del diccionario.

metricas_lista= glob.glob(os.path.join(r'Z:\LiDAR\Merge\*.tif'))
##lista de las imagenes de lidar

#####
##2.-EXTRACCIÓN DE MÉTRICAS:
#S1
print('Extrayendo métricas de LiDAR')
df5_copia = df5.copy() ##genero una copia de df5 --> lo deja intacto,
y luego se almacenará en df5_copia
#df50_pasada = df50.copy()

for i in metricas_lista:
    ncol = os.path.splitext(os.path.basename(i))[0] ##base4name me
coge el nombre de la ruta después de las dos últimas barras (\\). El
splittext me separa el dos el nombre justo donde el punto (merge.tif),
luego hacemos [0] para coger el primero.
    now = datetime.now()
    current_time = now.strftime("%H:%M:%S")
    print(ncol, current_time)
    img = rasterio.open(i) ##abrimos imagen.
    f5, c5 = img.index(x5,y5)
    img_values5 = pd.DataFrame(img.read(1)[f5, c5]) ##img.read me
abre la imagen como un dataframe.
    img_values5.columns = [ncol]
    df5_copia = pd.concat([df5_copia, img_values5], axis = 1)

print('extraccion acabada')
df5_copia.to_csv(os.path.join(dirfin, 'MuestreoHaya_05_LiDAR_bien.csv')
, index = False)
```

#### IV. ANEXO IV:

##### → Cálculo de métricas

```
from datetime import datetime
import geopandas as gpd
import rasterio
import pandas as pd
import glob
import os
import subprocess
```



```

from osgeo import gdal
import errno
import numpy as np
from shutil import rmtree
from os import remove
import shutil

## 1.- Directorios
dirLAStools = r'C:\LAStools\bin'
directorio = r'C:\XABIER\nubes_puntos_2'
#directorioNAS = r'T:\LiDAR\2017'

dirLAStools = r'C:\LAStools\bin'
dirfin = r'C:\XABIER\2017metricas5'
tamanopixel = 10
alturafcc = 3
eliminarArchivosfinales = 's' #S: no se elimina 'n': se elimina

#### 1.1. Listado de nubes de puntos a calcular
print('Leyendo nubes de puntos')
nubesdepuntos = glob.glob(os.path.join(directorio, '*.laz'))

##2.- FUNCIONES
def create_dir(ruta):
    try: os.mkdir(ruta)
    except OSError as e:
        if e.errno != errno.EEXIST: raise

def lasinfo_exe(dirinfoini, nombreinfo = '_info.txt', nombrecarpeta =
'1_Info', intensity = 100):
    LAStoolsexe = 'lasinfo'
    if os.path.exists(os.path.join(ncarpetaintermedia, nimagen +
nombreinfo)) == False:
        cmd = (dirLAStools + '\\ ' + LAStoolsexe + ' -i ' + dirinfoini
+ ' -histo intensity ' +str(intensity)
+ ' -odir ' + ncarpetaimagen + ' -o ' + nimagen +
nombreinfo)
        #print(cmd)
        subprocess.run(cmd)

def lastile_exe(dirtile, tiling = 250, cores = 8):
    global dirfintiles
    LAStoolsexe = 'lastile'
    create_dir(os.path.join(ncarpetaimagen, 'ArchivosIntermedios',
'tiles'))
    dirfintiles = os.path.join(ncarpetaimagen, 'ArchivosIntermedios',
'tiles')
    cmd = (dirLAStools + '\\ ' + LAStoolsexe + ' -i ' + dirtile
+ ' -tile_size ' +str(tiling)
+ ' -cores ' +str(cores)
+ ' -unindexed'
+ ' -odir ' +
os.path.join(ncarpetaimagen, 'ArchivosIntermedios', 'tiles') + ' -olaz')
    #print(cmd)
    subprocess.run(cmd)

def lasmerge_exe(dirinimerge, dirfinmerge):
    LAStoolsexe = 'lasmerge'
    cmd = (dirLAStools + '\\ ' + LAStoolsexe + ' -i ' + dirinimerge

```

```
        + ' -odir ' + dirfinmerge)
    ##print(cmd)
    subprocess.run(cmd)
def las2las_exe(dir2lasini, dropintensity = 1000, dropintensitytxt = '
-drop_intensity_below ', keep_class = ' -keep_class 2 3 4 5', drop_z =
'-drop_z_below 0', n2las = '_intensity_class_denoised.laz'):
    global dir2las, nombre2las
    nombre2las = n2las
    dir2las = os.path.join(ncarpetaimagen, nombre2las)
    LAsoolsexe = 'las2las'
    if os.path.exists(os.path.join(ncarpetaintermedia, nimagen +
n2las)) == False:
        cmd = (dirLAsools + '\\ ' + LAsoolsexe + ' -i ' + dir2lasini
            + dropintensitytxt + str(dropintensity)
            + keep_class
            + drop_z
            + ' -odir ' + ncarpetaintermedia + ' -o ' + nimagen +
n2las)
        ##print(cmd)
        subprocess.run(cmd)
def lasnoise_exe(dirnoiseini, nnoise = '_class2345_denoised.laz'):
    global nombrenoise
    nombrenoise = nnoise
    LAsoolsexe = 'lasnoise'
    if os.path.exists(os.path.join(ncarpetaintermedia, nimagen +
nombrenoise)) == False:
        cmd = (dirLAsools + '\\ ' + LAsoolsexe + ' -i ' + dirnoiseini
            + ' -step_xy 1.0 -step_z 0.2 -isolated 5 -
remove_noise'
            + ' -odir ' + ncarpetaintermedia + ' -o ' + nimagen +
nombrenoise)
        ##print(cmd)
        subprocess.run(cmd)
def las2dem_exe(dir2demini, pixel = tamanopixel, nombre2dem =
'_MDT.tif'):
    LAsoolsexe = 'blast2dem'
    cmd = (dirLAsools + '\\ ' + LAsoolsexe + ' -i ' + dir2demini
        + ' -step ' + str(pixel)
        + ' -keep_class 2'
        + ' -odir ' + ncarpetaimagen + ' -o ' + nimagen +
nombre2dem)
    #print(cmd)
    subprocess.run(cmd)
def lasheight_exe(dirhini, dirheightfinal, replace_z = '-replace_z',
ground_points = '', variable = ''):
    LAsoolsexe = 'lasheight'
    cmd = (dirLAsools + '\\ ' + LAsoolsexe + ' -i ' + dirhini
        + replace_z
        + ground_points
        + dirheightfinal)
    #print(cmd)
    subprocess.run(cmd)
def lasgrid_exe(dirgridini, metric, tipo, ncarpeta, variable = '',
pixel = tamanopixel, nombregrid = ''):
    LAsoolsexe = 'lasgrid'
    cmd = (dirLAsools + '\\ ' + LAsoolsexe + ' -i ' + dirgridini
        + ' -step ' + str(pixel)
```

```

        + metric #aquí iría la métrica que se quiera calcular (-
elevation, -intensity, etc.)
        + tipo #aquí se definiría el tipo de métrica: -average, -
max., -min., etc.
        + variable
        + '-odir ' + ncarpeta + '-o ' + nimagen + '_' +
nombregrid + '.tif')
        subprocess.run(cmd)
def lascanopy_exe(dircanopyini, metric, height_cutoff = '-
height_cutoff 0', variable = '', pixel = tamanopixel, nombrecanopy =
''):
    LAsoolsexe = 'lascanopy'
    cmd = (dirLAsools + '\\\ ' + LAsoolsexe + '-i ' + dircanopyini
        + '-step ' + str(pixel)
        + metric #se definiría como '-p 25' en el caso de
percentiles o '-ske' '/'-kur' o '-c 0 5 10 15 20 25'
        + height_cutoff
        + variable
        + '-odir ' + ncarpetaimagen + '-o ' + nimagen +
nombrecanopy + '.tif')
    subprocess.run(cmd)
def open_get_RasterBand(ruta, access = 0):
    img = gdal.Open(ruta, access)
    if img is None:
        print('Error al abrir el raster', ruta)
    band = img.GetRasterBand(1).ReadAsArray().astype(np.float32)
    return img, band
    img, band = None
def geotransform(imagen):
    global columnas, filas, proyeccion, geotrans
    geotrans = imagen.GetGeoTransform()
    Xorigen, anchura_pixel, vacio, Yfinal, vacio2, altura_pixel =
geotrans
    columnas = imagen.RasterXSize
    filas = imagen.RasterYSize
    proyeccion = imagen.GetProjection()
    return columnas, filas, proyeccion, geotrans
def saveRaster(dataset, datasetPath, columnas, filas, proyeccion,
geotrans):
    rasterSet = gdal.GetDriverByName('GTiff').Create(datasetPath,
columnas, filas, 1, gdal.GDT_Float32)
    rasterSet.SetProjection(proyeccion)
    rasterSet.SetGeoTransform(geotrans)
    rasterSet.GetRasterBand(1).WriteArray(dataset)
    rasterSet.GetRasterBand(1).SetNoDataValue(-999)
    rasterSet = None

pos0 = nubesdepuntos[0].rfind('600')
if pos0 < 0: pos0 = nubesdepuntos[0].rfind('_6')
nombretxt = nubesdepuntos[0][pos0:pos0+13]
file = open(os.path.join(dirfin, nombretxt + ".txt"), "w")

#####
#####
#3- Se aplican las funciones a las imagenes:
print('Empieza el proceso')
```

```
for directorio in nubesdepuntos:
    if nubesdepuntos.index(directorio) % 500 == 0:
print(nubesdepuntos.index(directorio), 'de', len(nubesdepuntos))
#Se genera una carpeta para cada imagen donde se iran guardando
los archivos (se genera también para archivos intermedios):
pos0 = directorio.rfind('\\')
pos1 = directorio.rfind('.')
nimagen0 = directorio[pos0+1:pos1]
x = 0
for i in range(len(nimagen0)):
    caracter = nimagen0[i]
    if caracter.isdigit() and x == 0:
        posicion = nimagen0.index(caracter)
        x = 1
nimagen = nimagen0[posicion:]
#print('Procesando nube de puntos', nimagen)
ncarpetaimagen = os.path.join(dirfin, nimagen)
ncarpetaintermedia = os.path.join(dirfin, nimagen,
'ArchivosIntermedios')
create_dir(ncarpetaimagen)
create_dir(ncarpetaintermedia)
####
#Informacion del producto inicial:
#lasinfo_exe(dirinfoini = directorio)
#Eliminar los puntos de intensidad baja y puntos que no
correspondan a las clases 2 3 4 5:
#las2las_exe(dir2lasini = directorio, drop_z = '')
#Eliminar ruido:
#lasnoise_exe(dirnoiseini = os.path.join(ncarpetaintermedia,
nimagen + '_intensity_class_denoised.laz'))
dirnoise = directorio
#if
os.stat(glob.glob(os.path.join(dirfin, nimagen, '*.txt'))[0]).st_size !=
0 and len(os.listdir(ncarpetaintermedia)) != 0:
    #if len(os.listdir(ncarpetaintermedia)) != 0:
    #dirNAS = dirnoise.replace(ncarpetaintermedia, directorioNAS)
    #if os.path.exists(dirNAS) == False: shutil.copy(dirnoise, dirNAS)
    #Generar MDT:
    #las2dem_exe(dir2demini = dirnoise)
    #Generar nube de puntos de alturas relativas:
    lasheight_exe(dirhini = dirnoise, dirheightfinal = ' -odir ' +
ncarpetaintermedia + ' -o ' + nimagen + '_nheight0.laz')
    #SI LA NUBE DE PUNTOS ES MUY GRANDE, LASHEIGHT DA ERROR, ASÍ QUE
HAY QUE SEPARAR EN TILES:
    if os.path.exists(os.path.join(ncarpetaintermedia, nimagen +
'_nheight0.laz')) == False:
        lastile_exe(dirtile = dirnoise)
        create_dir(os.path.join(dirfin, nimagen,
'ArchivosIntermedios', 'nheight'))
        lasheight_exe(dirhini = os.path.join(dirfintiles, '*laz'),
dirheightfinal = ' -odir ' + os.path.join(os.path.join(dirfin,
nimagen, 'ArchivosIntermedios', 'nheight')) + ' -olaz')
        lasmerge_exe(dirinimerge = os.path.join(os.path.join(dirfin,
nimagen, 'ArchivosIntermedios', 'nheight'), '*laz'), dirfinmerge =
ncarpetaintermedia + ' -o ' + nimagen + '_nheight0.laz')
        las2las_exe(dropintensity = '', dropintensitytxt = '', keep_class
= '', dir2lasini = ' -odir ' + os.path.join(ncarpetaintermedia, nimagen
```

```

+ '_nheight0.laz'), n2las = '_nheight.laz') #elimina puntos por debajo
de cota 0
    dir2las = os.path.join(ncarpetaintermedia, nimagen +
'_nheight.laz')
    #Generar rasters de interés a través de lasgrid:
    for metric in ['-elevation', '-intensity']:
        for tipo in ['-average', '-lowest', '-highest', '-std']:
            lasgrid_exe(dirgridini = dir2las, metric = metric,
tipo = tipo, ncarpeta = ncarpetaimagen, nombregrid = metric[2:] + '_'
+ tipo[2:])
            lasgrid_exe(dirgridini = dir2las, metric = '-point_density', tipo
= '', ncarpeta = ncarpetaimagen, nombregrid = 'density') #Densidad de
puntos
            lasgrid_exe(dirgridini = dir2las, metric = '-counter_32bit', tipo
= '', ncarpeta = ncarpetaimagen, nombregrid = 'npoints') #Número de
puntos por pixel
            lasgrid_exe(dirgridini = dir2las, metric = '-counter_32bit', tipo
= '', ncarpeta = ncarpetaimagen, variable = '-keep_return 1 -
drop_z_below ' + str(alturaafcc), nombregrid = 'fcc0_' +
str(alturaafcc)+'m')
            #fcclaz_img, fcclaz_band =
open_get_RasterBand(os.path.join(ncarpetaimagen,nimagen + '_fcc0_' +
str(alturaafcc)+'m.tif'))
            #npoints_img, npoints_band =
open_get_RasterBand(os.path.join(ncarpetaimagen,nimagen +
'_npoints.tif'))
            #if npoints_band.shape != fcclaz_band.shape:
                #file.write('Error del archivo'+directorio+'al generar el
fcc.')
            #else:
                #fcc = fcclaz_band / npoints_band
                #saveRaster(fcc, os.path.join(ncarpetaimagen,nimagen +
'_fcc_' + str(alturaafcc)+'m.tif'), columnas, filas, proyeccion,
geotrans)
            #fcclaz_img, fcclaz_band = None, None
            #Continuar generando rasters de interés a través de lascanopy:
            lascanopy_exe(dircanopyini = dir2las, metric = '-p 25 50 75 90
95', nombrecanopy = '_elevation')
            lascanopy_exe(dircanopyini = dir2las, metric = '-ske')
            lascanopy_exe(dircanopyini = dir2las, metric = '-kur')
            lascanopy_exe(dircanopyini = dir2las, metric = '-c 0 3 10 15 20
25 30 35', nombrecanopy = '_range')
            if eliminarArchivosfinales.lower() in ['s', 'si', 'sí']:
                rmtree(ncarpetaintermedia)
            remove(directorio)
file.close()
if os.stat("U:\LiDAR\Errores_" + nombretxt + ".txt").st_size == 0:
    remove("U:\LiDAR\Errores_" + nombretxt + ".txt")
    
```

## V. ANEXO V:

### → [Cálculo modelos de regresión](#)

```

import pandas as pd
import os
import sys
import glob
import numpy as np
    
```

```
from sklearn import metrics, model_selection
from sklearn.ensemble import RandomForestRegressor
import joblib
from datetime import datetime
from time import time, gmtime, strftime

CE = 0.70 #Dfinimos que porcentaje de muestras queremos en el conjunto
de entrenamiento(CE) (70/30)
dirfin =
r'C:\Users\goldaraz.112790\Desktop\Prubea_python\TFM\Regresion'

def nashsutcliffe(evaluation, simulation):
    s, e = np.array(simulation), np.array(evaluation)
    # s,e=simulation,evaluation
    mean_observed = np.nanmean(e)
    # compute numerator and denominator
    numerator = np.nansum((e - s) ** 2)
    denominator = np.nansum((e - mean_observed)**2)
    # compute coefficient
    return 1 - (numerator / denominator)

def RFRegressor(X_train, y_train, X_test, y_test):
    RF = RandomForestRegressor()
    param_grid = {'n_estimators': [10,50,100], 'max_features':
[1,'sqrt','log2',None], 'max_depth': [5,10,None], 'min_samples_split':
[2,10,20]}

    #Llamada la función GridSearchCV que nos crea todas las
cominaciones del grid anterior
    clasificadoresRF = model_selection.GridSearchCV(RF,
param_grid, cv=10)
    clasificadoresRF = clasificadoresRF.fit(X_train, y_train) ##Te
ajusta los parametros para tus datos de train.

    # Se muestra la mejor configuración y su accuracy asociado
    print(clasificadoresRF.best_params_)
    print(clasificadoresRF.best_score_)

    # LLamada al constructor del random forest utilizando los
parámetros del punto anterior
    bestparams = clasificadoresRF.best_params_ ##almacenamos los
mejores parametros.
    ##llamamos los mejores parametros para luego utilizarlos
    RF = RandomForestRegressor(max_depth =
bestparams['max_depth'],
                                max_features =
bestparams['max_features'],
                                min_samples_split =
bestparams['min_samples_split'],
                                n_estimators =
bestparams['n_estimators'])

    # Entrenamiento del árbol de decisión
    start_time = time()
    RF.fit(X_train, y_train) ##calcular el tiempo!
    elapsed_time = time() - start_time
    print(elapsed_time)
```

```
# joblib.dump(RF,os.path.join(dirfin,'RF_Regresion.joblib'),
compress = 3) --> Guarda el modelo ya entrenado

##PREDICCION
# Se realizan las predicciones del árbol de decisión con los
ejemplos de entrenamiento
predictionTrain = RF.predict(X_train)

# Se obtiene el rendimiento (accuracy)
r2train = metrics.r2_score(y_train, predictionTrain) #r2 de
entrenamiento.

msetrain = metrics.mean_squared_error(y_train,
predictionTrain)
RMSEtrain = np.sqrt(msetrain) ##RMS de entrenamiento.

NSEtrain = nashsutcliffe(y_train, predictionTrain) ##Otro
parametro como el RMS.

# Predicción de los datos de test
predictionTest = RF.predict(X_test)

# Cálculo del porcentaje de acierto en test (entre 0.0 y
100.0)
r2test = metrics.r2_score(y_test, predictionTest)
msetest = metrics.mean_squared_error(y_test, predictionTest)
RMSEtest = np.sqrt(msetest)
NSEtest = nashsutcliffe(y_test, predictionTest)
return r2train, RMSEtrain, NSEtrain, r2test, RMSEtest,
NSEtest, bestparams, predictionTrain, predictionTest, elapsed_time

## Graficos de las series predicha de cada variable con la
serie original de los datos de campo.
#Por eso almaceno los datos de predictionTrain y
predictionTest.--> las predicciones de train almaceno por precaución

directorio =
r'C:\Users\goldaraz.112790\Desktop\Prubea_python\TFM\regresiones_senti
nel2'
dflidar =
pd.read_csv(r'C:\Users\goldaraz.112790\Desktop\Prubea_python\TFM\Haya_
LiDAR_Sentinel2_bien.csv') #xlsx

variables = ['Altura_Dom','Area_Basim','Volumen_To'] #con el nombre de
las columnas
for variable in variables:
    print(variable)
    bandas = "LiDAR_Sentinel2"
    especie = "Haya"
    #Primero seleccionamos del df inicial la variable con la que se va
a trabajar
    colsint = [x for x in dflidar.columns if x not in variables]
    X = np.array(dflidar[colsint]) #Todas las metricas lidar mas X e y
    y = np.array(dflidar[variable])
```

```
##SEPARAR DATOS DE CALIBRACIÓN DE MODELO Y TEST MEDIANTE
VALIDACIÓN CRUZADA:
    print('Hold-out...') ## Separar los 70 para entrenar y 30 para
validar.
    np.random.seed(12) #Se fija la semilla de numpy para que la
generación aleatoria siempre nos de los mismos números
    X_train, X_test, y_train, y_test =
model_selection.train_test_split(X, y, train_size=CE, random_state=12)
    dftr, dftt = pd.DataFrame(X_train),pd.DataFrame(X_test) #Guardo en
modo dataframe, los conjuntos de entrenamiento de test y train (de las
variables predictoras). Se guardan con las coordenadas. El siguiente
paso entonces es eliminar esas puntos de las variables de regresion
# dftr --> dataframe de train dt dftt --> dataframe de test
dftr.columns = dftt.columns = colsint
colsxinxy = [x for x in colsint if x not in ['x','y']]
X_train = np.array(dftr[colsxinxy]) #todas las columnas de
metricas menos x e y.
X_test = np.array(dftt[colsxinxy])
    print('Ejecutando modelo...')
    print(strftime('%a,%d %b %Y %H:%M:%S', gmtime()))
    r2train, RMSEtrain, NSEtrain, r2test, RMSEtest, NSEtest, bparams,
yTrainpred, yTestpred, elapsed_time = RFRegressor(X_train, y_train,
X_test, y_test)

    YpredictTrain = pd.concat([dftr[['x','y']],
pd.DataFrame(yTrainpred)], axis = 1)
    YpredictTest = pd.concat([dftr[['x','y']],
pd.DataFrame(yTestpred)], axis = 1, join = 'inner')
    parametros = {
        'R2train': r2train,
        'RMSEtrain': RMSEtrain,
        'NSEtrain': NSEtrain,
        'R2test': r2test,
        'RMSEtest': RMSEtest,
        'NSEtest': NSEtest,
        'bparams': bparams,
        'elapsed_time': elapsed_time
    }

    df_parametros = pd.DataFrame(list(parametros.items()))
    df_parametros.to_csv(os.path.join(directorio,
bandas+especie+'_'+variable+'_parametros.csv'), index = False)
    YpredictTrain.to_csv(os.path.join(directorio,
bandas+especie+'_'+variable+'_YpredictTrain.csv'), index = True)
    YpredictTest.to_csv(os.path.join(directorio,
bandas+especie+'_'+variable+'_YpredictTest.csv'), index = True)
    print(strftime('%a,%d %b %Y %H:%M:%S', gmtime()))
```

## VI. ANEXO VI:

→ [Muestreo de F.Sylvatica:](#)

```
import pandas as pd
import geopandas as gpd
import os
```



```
points =
gpd.read_file('C:\\Users\\goldaraz.112790\\Desktop\\muestreo\\pruebas2
\\muestreoSentinel1.shp')
poligonos =
gpd.read_file('C:\\Users\\goldaraz.112790\\Desktop\\muestreo\\Pruebas\\
\\poligonostrabajo2.shp')

within_points = gpd.sjoin(points, poligonos, how='inner', op
='intersects')

0 in within_points['Area_Basim'].values
0 in points['SP1'].values

print(within_points.isnull().sum(axis=0))

# id_left          12
# SP1_left         12
# geometry         0
# index_right      0
# id_right         0
# IDPOM            0
# SP1_right        0
# Area_Basim       0
# Altura_Dom       0
# Volumen_To       0

within_points[0].value_counts()

dataHaya = within_points[within_points['SP1_right']== 213]

valoresnulos = within_points[within_points['SP1_right']== 'NaN']
print(dataHaya.isnull().sum(axis=0))

0 in dataHaya['Area_Basim'].values
0 in dataHaya['Volumen_To'].values
0 in dataHaya['SP1_left'].values
0 in dataHaya['Volumen_To'].values
0 in dataHaya['SP1_right'].values
0 in dataHaya['Area_Basim'].values
0 in dataHaya['Area_Basim'].values
0 in dataHaya['Area_Basim'].values

#####
#####
## HACER CUARTILES
#####
#####

qs_Altura = dataHaya.Altura_Dom.quantile([.25, .5, .75])
# 0.25    18.0
# 0.50    22.0
# 0.75    26.0
```

```
qs_Volumen = dataHaya.Volumen_To.quantile([.25, .5, .75])
# 0.25    135.660000
# 0.50    222.889606
# 0.75    305.760000

qs_Area = dataHaya.Area_Basim.quantile([.25, .5, .75])
# 0.25    19.0
# 0.50    25.0
# 0.75    31.0

##MUESTREO DEL 5% DEL HAYA
# dataHaya = dataHaya.sample(frac = 0.05) ##44.444 puntos de muestreo

###ASIGNAR LOS CUARTILES A LOS DATOS
dataHaya['qs_Altura'] = pd.qcut(dataHaya['Altura_Dom'], q= 4,
labels=False)

dataHaya['qs_Volumen'] = pd.qcut(dataHaya['Volumen_To'], q= 4,
labels=False)

dataHaya['qs_Area'] = pd.qcut(dataHaya['Area_Basim'], q= 4,
labels=False)

directorio =
r'C:\\Users\\goldaraz.112790\\Desktop\\muestreo\\muestreo_hayas'

datoscuartiles = dataHaya[['Area_Basim', 'Altura_Dom', 'Volumen_To',
'qs_Altura', 'qs_Volumen', 'qs_Area', 'geometry']]

dataHaya_1 = datoscuartiles[(datoscuartiles['qs_Volumen'] ==
0)].sample(frac = 0.05)
dataHaya_2 = datoscuartiles[(datoscuartiles['qs_Volumen'] ==
1)].sample(frac = 0.05)
dataHaya_3 = datoscuartiles[(datoscuartiles['qs_Volumen'] ==
2)].sample(frac = 0.05)
dataHaya_4 = datoscuartiles[(datoscuartiles['qs_Volumen'] ==
3)].sample(frac = 0.05)

frames = [dataHaya_1, dataHaya_2, dataHaya_3, dataHaya_4]

data0 = pd.concat(frames)

data0.to_file(os.path.join(directorio,
'DataHayaMuestreo_05_bien.shp'), driver = 'ESRI Shapefile')
```

## VII. ANEXO VII:

→ [Compuestos anuales de Sentinel-1 y Sentinel-2:](#)

```
import os
import glob
from osgeo import gdal
import errno
import numpy as np
```

```

from datetime import datetime
from datetime import date

aviso = True
#DEFINICION DE FUNCIONES
def create_dir(ruta):
    try: os.mkdir(ruta)
    except OSError as e:
        if e.errno != errno.EEXIST: raise
def tiempo():
    today = str(date.today())
    now = datetime.now()
    current_time = now.strftime("%H:%M:%S")
    return current_time+' '+today
def open_Raster(ruta, access = 0):
    img = gdal.Open(ruta, access)
    if img is None:
        print('Error al abrir el raster',ruta)
    return img
    img = None
def get_RasterBand(img, access = 0):
    band = img.GetRasterBand(1).ReadAsArray().astype(np.float32)
    band[band > 10000] = np.quantile(band, 0.99)
    band = np.array(band).astype('float')
    band[band==0] = np.nan
    return band
    band = None
def geotransform(imagen):
    geotrans = imagen.GetGeoTransform()
    Xorigen, anchura_pixel, vacio, Yfinal, vacio2, altura_pixel =
geotrans
    columnas = imagen.RasterXSize
    filas = imagen.RasterYSize
    proyeccion = imagen.GetProjection()
    return columnas, filas, proyeccion, geotrans
def saveRaster(dataset, datasetPath, columnas, filas, proyeccion,
geotrans):
    rasterSet = gdal.GetDriverByName('GTiff').Create(datasetPath,
columnas, filas, 1, gdal.GDT_Float32)
    rasterSet.SetProjection(proyeccion)
    rasterSet.SetGeoTransform(geotrans)
    rasterSet.GetRasterBand(1).WriteArray(dataset)
    rasterSet.GetRasterBand(1).SetNoDataValue(np.nan)
    rasterSet = None

#EJECUCION DEL PROGRAMA (llamada a las funciones)
bandasS2 =
['B02', 'B03', 'B04', 'B08', 'B05', 'B06', 'B07', 'B11', 'B12', 'NDVI',
'NDII', 'NBR']
directorio = r'Z:\Imágenes_satelite\Optico\Sentinel-2\2018\Mensual'
directoriofinal = r''
for bandaS2 in bandasS2:
    if aviso: print('Banda',bandaS2)
    bandax =
glob.glob(os.path.join(directorio, '*', '*'+bandaS2+'.tif'))
    bandaxnpix =
glob.glob(os.path.join(directorio, '*', '*'+bandaS2+'_npix.tif'))
    
```

```
listasumas,listasumaspix = [], []
for i in bandax:
    array_listx=get_RasterBand(open_Raster(i))
    suma0x = np.nansum(array_listx, axis = 0)
    listasumas.append(suma0x)
for x in bandaxnpix:
    array_listxnpix=get_RasterBand(open_Raster(x))
    suma0xnpix = np.nansum(array_listxnpix, axis = 0)
    listasumaspix.append(suma0xnpix)
print('empezando suma final..' + bandaS2)
sumax = np.nansum(listasumas, axis = 0)
sumaxnpix = np.nansum(listasumaspix, axis = 0)
anual = sumax/sumaxnpix
dirfinal = r'C:\Users\goldaraz.112790\Desktop\Compuestos_Anuales'
nombrefin = 'Compuesto_Anual_'
columnas, filas, proyeccion, geotrans =
geotransform(open_Raster(bandax[0]))
saveRaster(anual, os.path.join(dirfinal,nombrefin+bandaS2),
columnas, filas, proyeccion, geotrans)
```

## VIII. ANEXO VIII:

→ [Descarga y preproceso LiDAR:](#)

```
import urllib, urllib.request
import os
import errno

url0 = 'https://filescartografia.navarra.es/5_LIDAR/'
with urllib.request.urlopen(url0) as html_handler0:
    html0 = str(html_handler0.read(), 'utf-8')
lista0 = html0.split(sep = '<br>')
lista = [linea for linea in lista0 if '&lt;dir&gt;' in linea]
listadoaños = ['2011', '2012', '2017']
#Solicitamos el año de descarga:
añodescarga0 = [x for x in input('Introduce el/los año(s) a descargar
(2011, 2012, 2017)\n(si se quiere más de un año, separar mediante
espacio): \n').split()]
#Comprobamos que los años introducidos son válidos:
comparacion = [item for item in añodescarga0 if item in listadoaños]
while len(añodescarga0) > 3 or len(comparacion) != len(añodescarga0):
    añodescarga0 = [x for x in input('Valores incorrectos.\nIntroduce
el/los año(s) a descargar (2011, 2012, 2017)\n(si se quiere más de un
año, separar mediante espacio): \n').split()]
    comparacion = [item for item in añodescarga0 if item in
listadoaños]
lista0 = []
for añodescarga in añodescarga0:
    #Una vez sabemos el año que queremos descargar, tenemos que
comprobar que, si es 2012, sea zona norte o sur:
    if añodescarga == '2012':
        zona = input('Indica la zona que quiera descargar del año 2012
(NORTE (N/n), SUR (S/s) o ambas): ')
        while zona.lower() not in ['n','norte','s','sur', 'ambas']:
            zona = input('Zona no disponible. Vuelve a indicar la zona
del año 2012 (NORTE (N/n), SUR (S/s) o ambas): ')

```

```

        zona = 'NORTE' if zona.lower() == 'n' or zona.lower() ==
'norte' else 'SUR' if zona.lower() == 's' or zona.lower() == 'sur'
else 'ambas'
        #Nos quedamos con la línea de la que nos interese el año de
descarga:
        for linea in lista:
            if añodescarga != '2012':
                if añodescarga in linea:
                    l = linea.split(sep = '<A HREF="/5_LIDAR/')
                    lista0.append(l[1].split(sep = '>')[0])
                else:
                    if añodescarga in linea and zona.lower() == 'norte' and
'NORTE' in linea:
                        l = linea.split(sep = '<A HREF="/5_LIDAR/')
                        lista0.append(l[1].split(sep = '>')[0])
                    elif añodescarga in linea and zona.lower() == 'sur' and
'SUR' in linea:
                        l = linea.split(sep = '<A HREF="/5_LIDAR/')
                        lista0.append(l[1].split(sep = '>')[0])
                    else:
                        if añodescarga in linea and zona.lower() == 'ambas':
                            if 'NORTE' in linea:
                                l = linea.split(sep = '<A HREF="/5_LIDAR/')
                                lista0.append(l[1].split(sep = '>')[0])
                            if 'SUR' in linea:
                                l = linea.split(sep = '<A HREF="/5_LIDAR/')
                                lista0.append(l[1].split(sep = '>')[0])
dirfinal = input('Introduce el directorio donde desee guardar la
carpeta con los archivos (se generará una carpeta con el nombre del
año donde irán las descargas):')
for lista00 in lista0:
    posicion0 = lista00.find('_NAV')
    añodescarga = lista00[posicion0-4:posicion0]
    #Unimos para obtener el url del año de interés:
    url = url0 + lista00
    with urllib.request.urlopen(url) as html_handler:
        html = str(html_handler.read(), 'utf-8')
    #Generamos una lista con las líneas que nos interesan:
    listado0 = html.split(sep = '<A HREF=')
    listado00 = [line for line in listado0 if '.laz' in line]
    lista2 = []
    for line in listado00: lista2.append(line.split(sep =
'>las_cm')[0])
    #Aquí podrían seleccionarse solo las hojas que nos interesaran, y
no TODAS las que están disponibles, en función de las coordenadas:
    lista3 = []
    for x in lista2:
        pos1 = x.find('.laz')
        nombrelaz0 = x[:pos1+4]
        pos0 = nombrelaz0.rfind(r'/')
        nombrelaz = nombrelaz0[pos0+1:]
        lista3.append(url+nombrelaz)

##DESCARGA DE NUBES DE PUNTOS:
dirnuevacarpeta = os.path.join(dirfinal,str(añodescarga))
try:
    os.mkdir(dirnuevacarpeta)

```

```
except OSError as e:
    if e.errno != errno.EEXIST:
        raise
    contenidolista3 = os.listdir(dirnuevacarpeta)
    print('Descargando nubes de puntos del año', año, aiodescarga)
    if len(contenidolista3) != 0: lista3 =
lista3[len(contenidolista3):] #Esto es por si ya hay alguna nube de
puntos descargada.
    #for i in range(5): #PARA DESCARGAR LAS 5 PRIMERAS. SI SE QUIERE
TODAS, DESACTIVAR ESTO Y ACTIVAR LA SIGUIENTE LÍNEA
    for i in range(len(lista3)): #PARA DESCARGAR TODO
        if i % 10 == 0:
print('Descargando', str(i), '/', str(len(lista3)))
        nombrefinal = lista3[i][lista3[i].rfind(r'/')+1:]
        urllib.request.urlretrieve(lista3[i],
os.path.join(dirnuevacarpeta, nombrefinal))
        if len(os.listdir(dirnuevacarpeta)) > 300:
            exec(open("Preproceso_LiDAR.py").read())
```

```
import urllib
import urllib.request
import pandas as pd
import os
import errno
import sys
sys.path.insert(0, r'D:\forestOBS\Scripts')
from Preproceso_LiDAR_unidoadescarga_vv import preproceso_LiDAR
sys.path.insert(0, r'D:\forestOBS\20-XABI') #directorio: ruta completa
del directorio donde se enucentre el script funciones. Por ejemplo:
r'D:\Funcionespy'
from directorios_faltan_calcular import dirfalta

dirfinal = r'T:\LiDAR\2017' #Esto en principio estaba como input. Lo
he dejado así para hacerlo ahora de forma automática.
dirmetricas = r'T:\LiDAR\2017_metricas2' #Esto también podría ponerse
como input.

my_file = open("T:\LiDAR\LinksLiDAR.txt", "r")
content = my_file.read()
content_list = content.split("\n")
my_file.close()
#carpetas = [x for x in os.listdir(dirfinal) if 'sh' not in x]
#falta = [x for x in carpetas if
len(os.listdir(os.path.join(dirfinal,x))) == 0]
falta, proceso = dirfalta(dirdata =
r'T:\LiDAR\Falta_sincv2fccMDT.xlsx')
for file in proceso: preproceso_LiDAR(dir0 = file, dirfin =
dirmetricas)

descarga = [x for x in content_list if
x[x.rfind('cam_')+4:x.rfind('cam_')+7] in falta]
for link in descarga: #PARA DESCARGAR TODO
    ncarpeta = link[link.rfind('cam_')+4:link.rfind('cam_')+7]
    nombrefinal = os.path.basename(link)
    urllib.request.urlretrieve(link,
os.path.join(dirfinal,ncarpeta,nombrefinal))
```

```

preproceso_LiDAR(dir0 =
os.path.join(dirfinal,ncarpeta,nombrefinal), dirfin = dirmetricas)
    #print('Descargando',os.path.join(dirnuevacarpeta,nombrefinal))

falta2 = pd.read_excel(r'T:\LiDAR\RutasFalta.xlsx')
list0 = list(falta2['elevation_average'].dropna())
listaproceso = [os.path.dirname(x)+'_class2345_denoised.laz' for x in
list0]
listaproceso = [x.replace('_metricas2','') for x in listaproceso]
INTERES = [os.path.basename(x)[:8] for x in listaproceso]
for file in listaproceso: preproceso_LiDAR(dir0 = file, dirfin =
dirmetricas)
    
```

```

def preproceso_LiDAR(dirLASTools = r'C:\LASTools\bin', dir0 =
r'U:\LiDAR\2017', dirfin = r'U:\LiDAR\2017_metricas', tamanopixel =
10, alturafcc = 3, eliminarArchivosfinales = 's', directorioNAS =
r'T:\LiDAR\2017'):
    import os
    import subprocess
    import glob
    from osgeo import gdal
    import errno
    import numpy as np
    from shutil import rmtree
    from os import remove
    import shutil
    #Se definen funciones con los .exe de LASTools que se van a
    utilizar
    def create_dir(ruta):
        try: os.mkdir(ruta)
        except OSError as e:
            if e.errno != errno.EEXIST: raise
    def lasinfo_exe(dirinfoini, nombreinfo = '_info.txt',
nombrecarpeta = '1_Info', intensity = 100):
        LASToolsexe = 'lasinfo'
        if os.path.exists(os.path.join(ncarpetaintermedia, nimagen +
nombreinfo)) == False:
            cmd = (dirLASTools + '\\\' + LASToolsexe + ' -i ' +
dirinfoini
                    + ' -histo intensity ' +str(intensity)
                    + ' -odir ' + ncarpetaimagen + ' -o ' + nimagen +
nombreinfo)
            ##print(cmd)
            subprocess.run(cmd)
    def lastile_exe(dirtile, tiling = 500, cores = 8):
        global dirfintiles
        LASToolsexe = 'lastile'
        create_dir(os.path.join(ncarpetaimagen, 'ArchivosIntermedios',
'tiles'))
        dirfintiles =
os.path.join(ncarpetaimagen, 'ArchivosIntermedios', 'tiles')
        cmd = (dirLASTools + '\\\' + LASToolsexe + ' -i ' + dirstile
                + ' -tile_size ' +str(tiling)
                + ' -cores ' +str(cores)
                + ' -unindexed'
    
```

```

        + ' -odir ' +
os.path.join(ncarpetaimagen, 'ArchivosIntermedios', 'tiles') + ' -olaz')
#print(cmd)
subprocess.run(cmd)
def lasmerge_exe(dirinimerge, dirfinmerge):
    LAsoolsexe = 'lasmerge'
    cmd = (dirLAsools + '\\ ' + LAsoolsexe + ' -i ' + dirinimerge
          + ' -odir ' + dirfinmerge)
    ##print(cmd)
    subprocess.run(cmd)
def las2las_exe(dir2lasini, dropintensity = 1000, dropintensitytxt
= ' -drop_intensity_below ', keep_class = ' -keep_class 2 3 4 5',
drop_z = ' -drop_z_below 0', n2las = ' _intensity_class_denoised.laz'):
    global dir2las, nombre2las
    nombre2las = n2las
    dir2las = os.path.join(ncarpetaimagen, nombre2las)
    LAsoolsexe = 'las2las'
    if os.path.exists(os.path.join(ncarpetaintermedia, nimagen +
n2las)) == False:
        cmd = (dirLAsools + '\\ ' + LAsoolsexe + ' -i ' +
dir2lasini
                + dropintensitytxt + str(dropintensity)
                + keep_class
                + drop_z
                + ' -odir ' + ncarpetaintermedia + ' -o ' +
nimagen + n2las)
        ##print(cmd)
        subprocess.run(cmd)
def lasnoise_exe(dirnoiseini, nnoise =
'_class2345_denoised.laz'):
    global nombrenoise
    nombrenoise = nnoise
    LAsoolsexe = 'lasnoise'
    if os.path.exists(os.path.join(ncarpetaintermedia, nimagen +
nombrenoise)) == False:
        cmd = (dirLAsools + '\\ ' + LAsoolsexe + ' -i ' +
dirnoiseini
                + ' -step_xy 1.0 -step_z 0.2 -isolated 5 -
remove_noise'
                + ' -odir ' + ncarpetaintermedia + ' -o ' +
nimagen + nombrenoise)
        ##print(cmd)
        subprocess.run(cmd)
def las2dem_exe(dir2demini, pixel = tamanopixel, nombre2dem =
'_MDT.tif'):
    LAsoolsexe = 'blast2dem'
    cmd = (dirLAsools + '\\ ' + LAsoolsexe + ' -i ' + dir2demini
          + ' -step ' + str(pixel)
          + ' -keep_class 2'
          + ' -odir ' + ncarpetaimagen + ' -o ' + nimagen +
nombre2dem)
    #print(cmd)
    subprocess.run(cmd)
def lasheight_exe(dirhini, dirheightfinal, replace_z = ' -
replace_z', ground_points = '', variable = ''):
    LAsoolsexe = 'lasheight'
    cmd = (dirLAsools + '\\ ' + LAsoolsexe + ' -i ' + dirhini

```



```

        + replace_z
        + ground_points
        + dirheightfinal)
    #print(cmd)
    subprocess.run(cmd)
    def lasgrid_exe(dirgridini, metric, tipo, ncarpeta, variable = '',
pixel = tamanopixel, nombregrid = ''):
        LAAstoolsexe = 'lasgrid'
        cmd = (dirLAAstools + '\\\ ' + LAAstoolsexe + ' -i ' + dirgridini
            + ' -step ' + str(pixel)
            + metric #aquí iría la métrica que se quiera calcular
(-elevation, -intensity, etc.)
            + tipo #aquí se definiría el tipo de métrica: -
average, -max., -min., etc.
            + variable
            + ' -odir ' + ncarpeta + ' -o ' + nimagen + '_' +
nombregrid + '.tif')
        subprocess.run(cmd)
    def lascanopy_exe(dircanopyini, metric, height_cutoff = '-
height_cutoff 0', variable = '', pixel = tamanopixel, nombrecanopy =
''):
        LAAstoolsexe = 'lascanopy'
        cmd = (dirLAAstools + '\\\ ' + LAAstoolsexe + ' -i ' +
dircanopyini
            + ' -step ' + str(pixel)
            + metric #se definiría como ' -p 25' en el caso de
percentiles o ' -ske' /' -kur' o '-c 0 5 10 15 20 25'
            + height_cutoff
            + variable
            + ' -odir ' + ncarpetaimagen + ' -o ' + nimagen +
nombrecanopy + '.tif')
        subprocess.run(cmd)
    def open_get_RasterBand(ruta, access = 0):
        img = gdal.Open(ruta, access)
        if img is None:
            print('Error al abrir el raster', ruta)
        band = img.GetRasterBand(1).ReadAsArray().astype(np.float32)
        return img, band
        img, band = None
    def geotransform(imagen):
        global columnas, filas, proyeccion, geotrans
        geotrans = imagen.GetGeoTransform()
        Xorigen, anchura_pixel, vacio, Yfinal, vacio2, altura_pixel =
geotrans
        columnas = imagen.RasterXSize
        filas = imagen.RasterYSize
        proyeccion = imagen.GetProjection()
        return columnas, filas, proyeccion, geotrans
    def saveRaster(dataset, datasetPath, columnas, filas, proyeccion,
geotrans):
        rasterSet = gdal.GetDriverByName('GTiff').Create(datasetPath,
columnas, filas, 1, gdal.GDT_Float32)
        rasterSet.SetProjection(proyeccion)
        rasterSet.SetGeoTransform(geotrans)
        rasterSet.GetRasterBand(1).WriteArray(dataset)
        rasterSet.GetRasterBand(1).SetNoDataValue(-999)
        rasterSet = None

```

```
directorio = dir0
#Se genera una carpeta para cada imagen donde se iran guardando
los archivos (se genera también para archivos intermedios):
nimagen,_ = os.path.basename(directorio).split('.')
nimagen = nimagen.replace('las_cam_', '') if 'las_cam_' in nimagen
else nimagen
#print('Procesando nube de puntos', nimagen)
ncarpetaimagen = os.path.join(dirfin, nimagen)
ncarpetaintermedia = os.path.join(dirfin, nimagen,
'ArchivosIntermedios')
create_dir(ncarpetaimagen)
create_dir(ncarpetaintermedia)
####
#Informacion del producto inicial:
lasinfo_exe(dirinfoini = directorio)
#Eliminar los puntos de intensidad baja y puntos que no
correspondan a las clases 2 3 4 5:
las2las_exe(dir2lasini = directorio, drop_z = '')
#Eliminar ruido:
lasnoise_exe(dirnoiseini = os.path.join(ncarpetaintermedia,
nimagen + '_intensity_class_denoised.laz'))
dirnoise = os.path.join(ncarpetaintermedia, nimagen + nombrenoise)

shutil.copy(dirnoise, os.path.join(directorio[:directorio.rfind(os.path
.basename(directorio))], nimagen + nombrenoise))
if
os.stat(glob.glob(os.path.join(dirfin, nimagen, '*.txt'))[0]).st_size !=
0 and len(os.listdir(ncarpetaintermedia)) != 0:
#dirNAS = dirnoise.replace(ncarpetaintermedia, directorioNAS)
#if os.path.exists(dirNAS) == False: shutil.copy(dirnoise,
dirNAS)
#Generar MDT:
las2dem_exe(dir2demini = dirnoise)
#Generar nube de puntos de alturas relativas:
lasheight_exe(dirhini = dirnoise, dirheightfinal = '-odir ' +
ncarpetaintermedia + '-o ' + nimagen + '_nheight0.laz')
#SI LA NUBE DE PUNTOS ES MUY GRANDE, LASHEIGHT DA ERROR, ASÍ
QUE HAY QUE SEPARAR EN TILES:
if os.path.exists(os.path.join(ncarpetaintermedia, nimagen +
'_nheight0.laz')) == False:
lastile_exe(dirtile = dirnoise)
create_dir(os.path.join(dirfin, nimagen,
'ArchivosIntermedios', 'nheight'))
lasheight_exe(dirhini = os.path.join(dirfintiles, '*laz'),
dirheightfinal = '-odir ' + os.path.join(os.path.join(dirfin,
nimagen, 'ArchivosIntermedios', 'nheight')) + '-olaz')
lasmerge_exe(dirinimerge =
os.path.join(os.path.join(dirfin, nimagen,
'ArchivosIntermedios', 'nheight'), '*laz'), dirfinmerge =
ncarpetaintermedia + '-o ' + nimagen + '_nheight0.laz')
las2las_exe(dropintensity = '', dropintensitytxt = '',
keep_class = '', dir2lasini = '-odir ' +
os.path.join(ncarpetaintermedia, nimagen + '_nheight0.laz'), n2las =
'_nheight.laz') #elimina puntos por debajo de cota 0
dir2las = os.path.join(ncarpetaintermedia, nimagen +
'_nheight.laz')
```

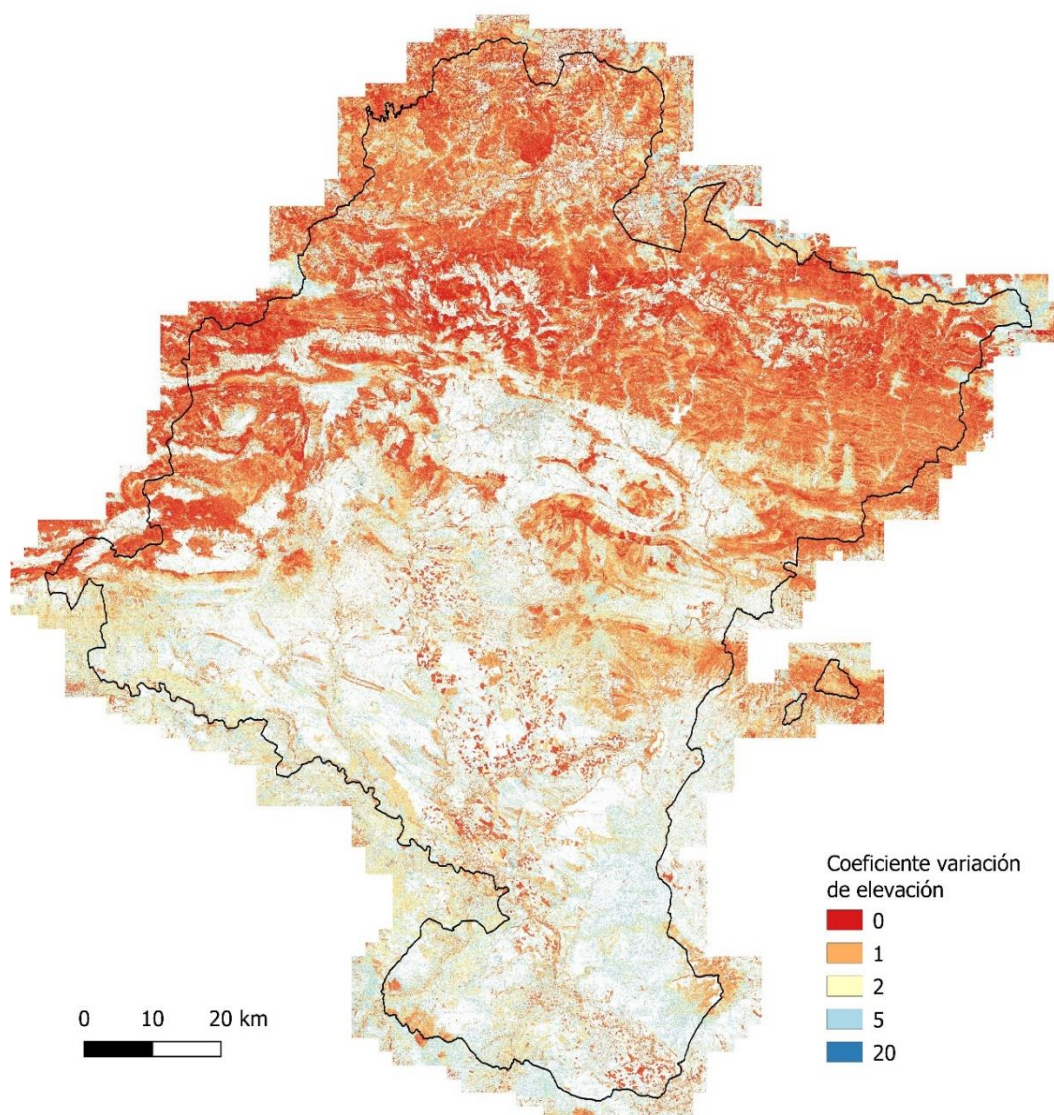
```

shutil.copy(dir2las,os.path.join(directorio[:directorio.rfind(os.path.
basename(directorio))],nimagen + '_nheight.laz'))
#Generar rasters de interés a través de lasgrid:
for metric in ['-elevation', '-intensity']:
    for tipo in ['-average', '-lowest', '-highest', '-
std']:
        lasgrid_exe(dirgridini = dir2las, metric = metric,
tipo = tipo, ncarpeta = ncarpetaimagen, nombregrid = metric[2:] + '_'
+ tipo[2:])
        lasgrid_exe(dirgridini = dir2las, metric = '-point_density',
tipo = '', ncarpeta = ncarpetaimagen, nombregrid = 'density')
#Densidad de puntos
        lasgrid_exe(dirgridini = dir2las, metric = '-counter_32bit',
tipo = '', ncarpeta = ncarpetaimagen, nombregrid = 'npoints') #Número
de puntos por pixel
        lasgrid_exe(dirgridini = dir2las, metric = '-counter_32bit',
tipo = '', ncarpeta = ncarpetaintermedia, variable = '-keep_return 1
-drop_z_below ' + str(alturafcc), nombregrid = 'fcc0_' +
str(alturafcc)+'m')
        fcclaz_img, fcclaz_band =
open_get_RasterBand(os.path.join(ncarpetaintermedia,nimagen +
'_fcc0_' + str(alturafcc)+'m.tif'))
        npoints_img, npoints_band =
open_get_RasterBand(os.path.join(ncarpetaimagen,nimagen +
'_npoints.tif'))
        try:
            fcc = fcclaz_band / npoints_band
            columnas, filas, proyeccion, geotrans =
geotransform(fcclaz_img)
            saveRaster(fcc, os.path.join(ncarpetaimagen,nimagen +
'_fcc_' + str(alturafcc)+'m.tif'), columnas, filas, proyeccion,
geotrans)
            fcclaz_img, fcclaz_band = None, None
        except:
            print('fcc3m no calculada')
            fcclaz_img, fcclaz_band = None, None
#Continuar generando rasters de interés a través de lascanopy:
lascanopy_exe(dircanopyini = dir2las, metric = '-p 25 50 75
90 95', nombrecanopy = '_elevation')
lascanopy_exe(dircanopyini = dir2las, metric = '-ske')
lascanopy_exe(dircanopyini = dir2las, metric = '-kur')
lascanopy_exe(dircanopyini = dir2las, metric = '-c 0 3 10 15
20 25 30 35', nombrecanopy = '_range')
lascanopy_exe(dircanopyini = dir2las, metric = '-dns',
height_cutoff='-height_cutoff '+str(alturafcc), nombrecanopy='_fcc')
if eliminarArchivosfinales.lower() in ['s', 'si','sí']:
    rmtree(ncarpetaintermedia)
#remove(directorio)

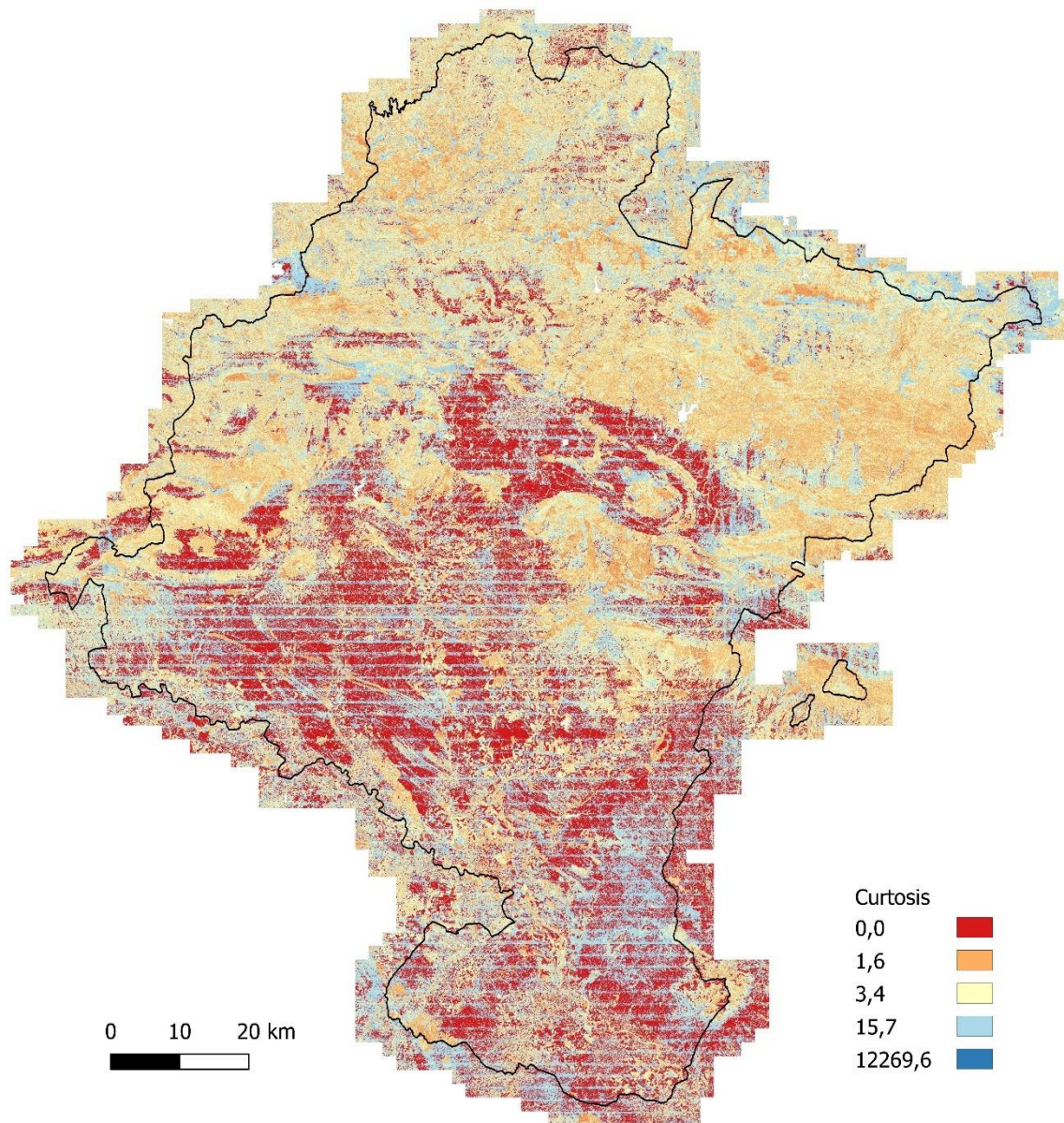
```

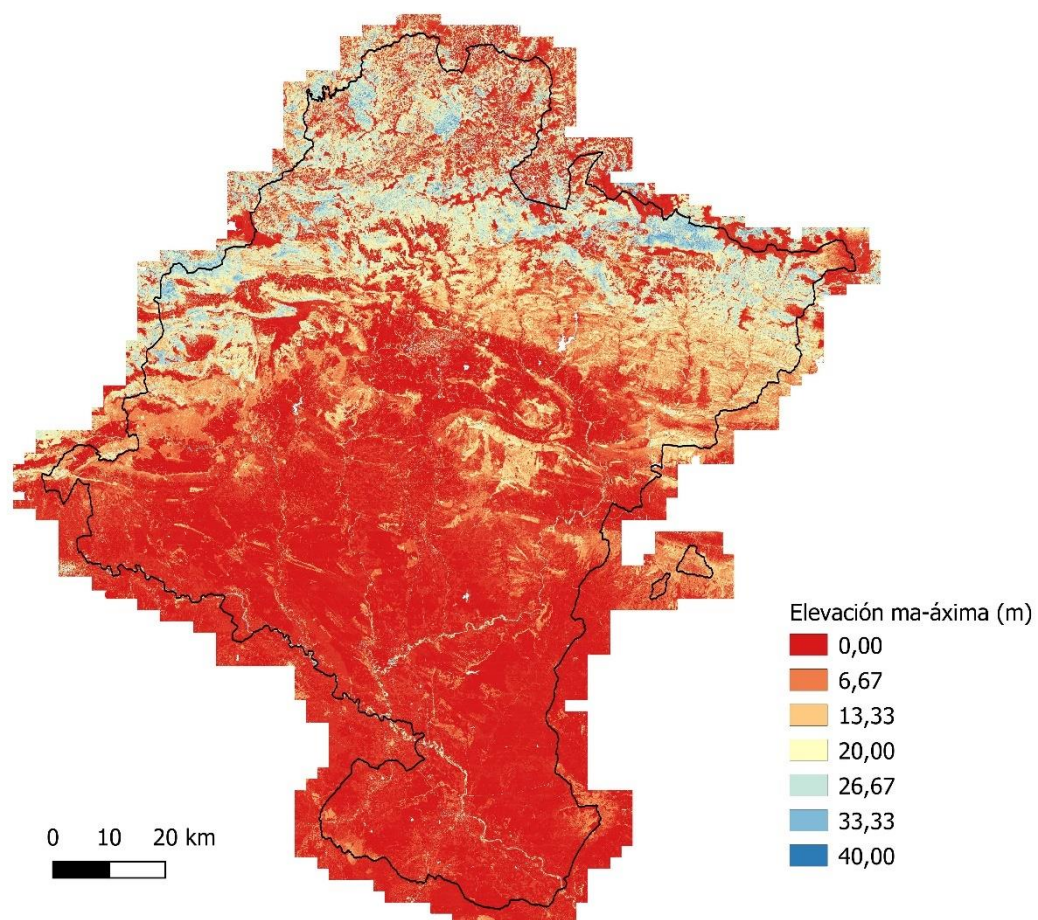
## IX. ANEXO IX:

→ [Mapas de los mosaicos de las métricas LiDAR:](#)

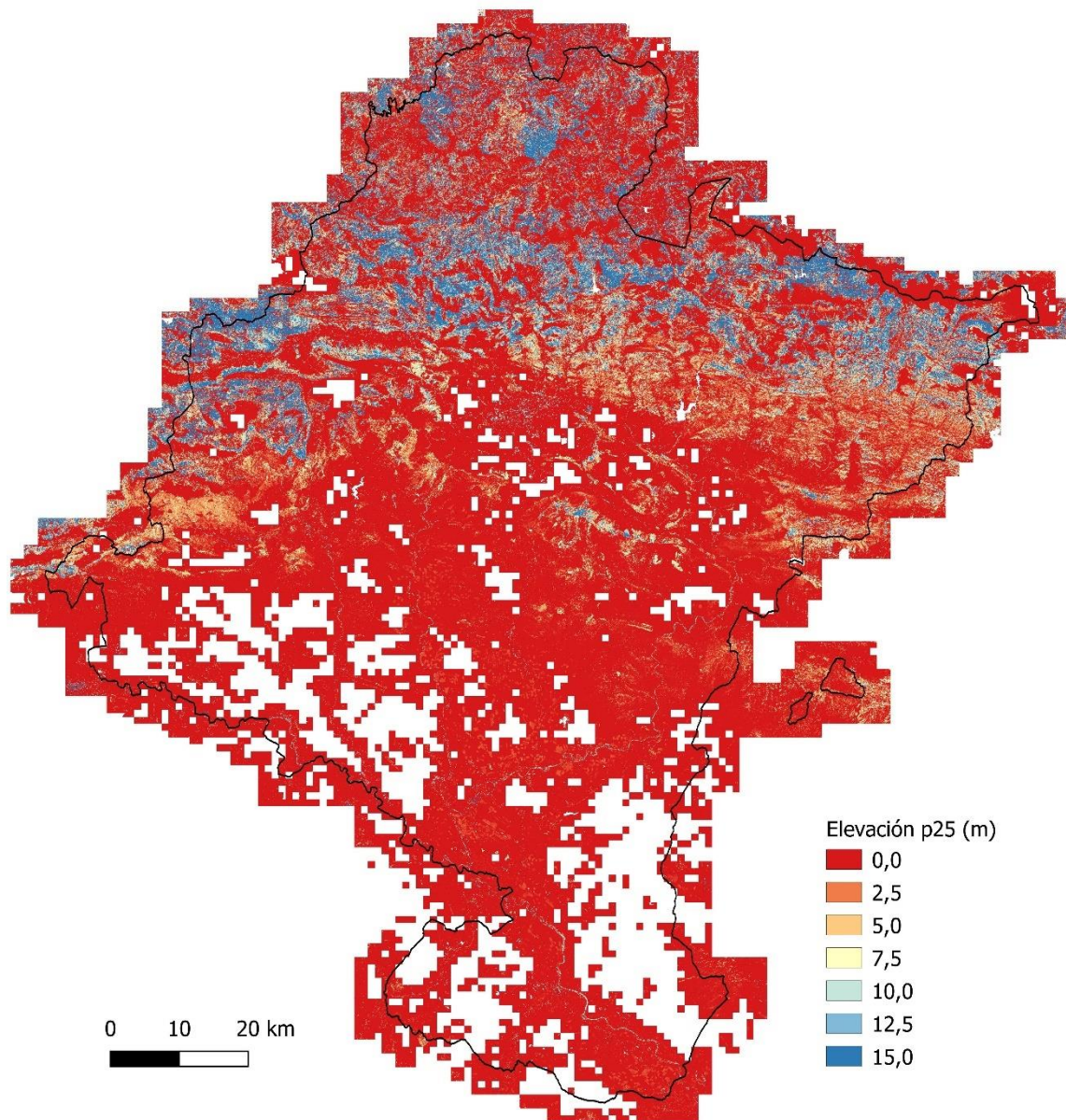


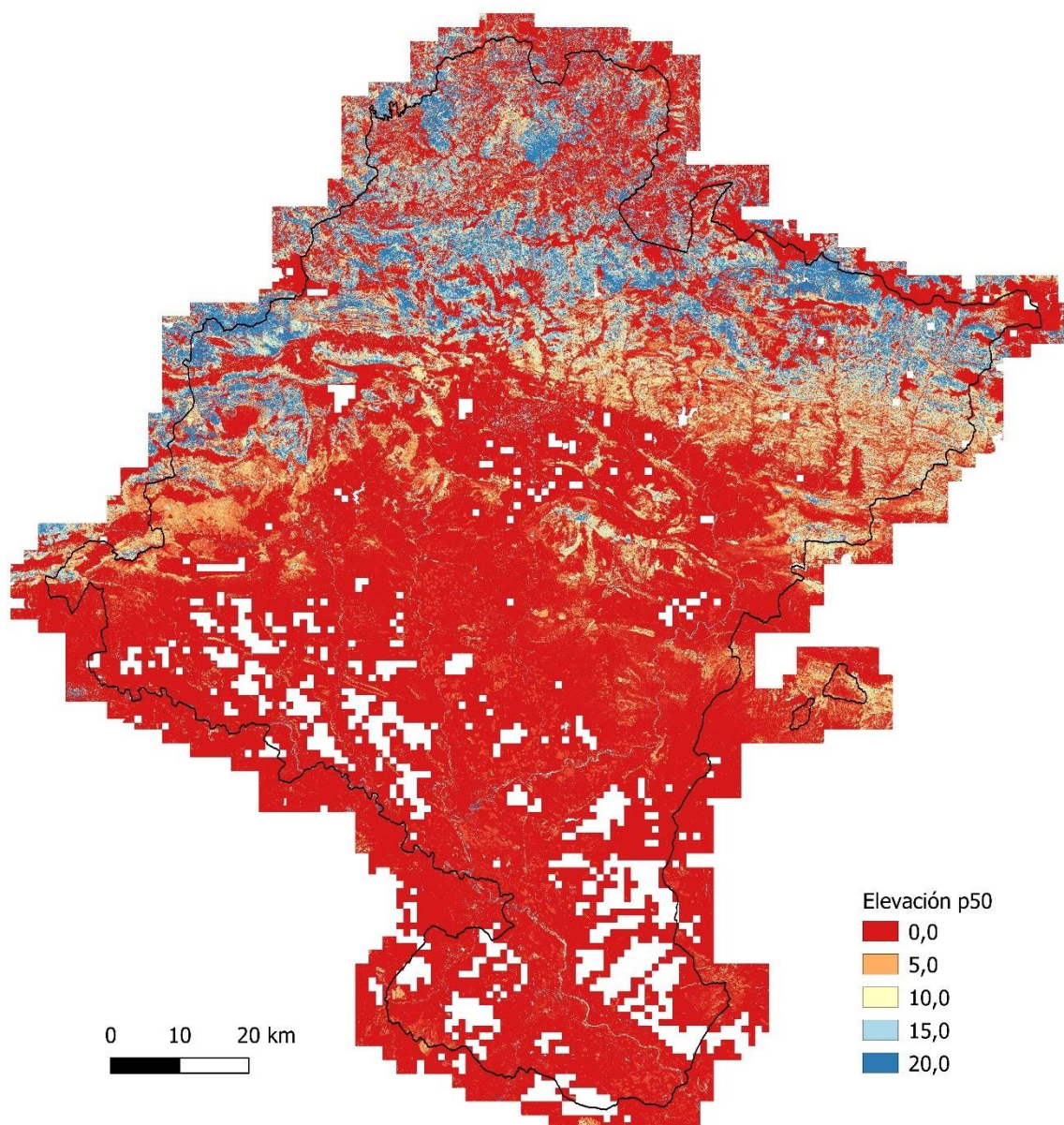




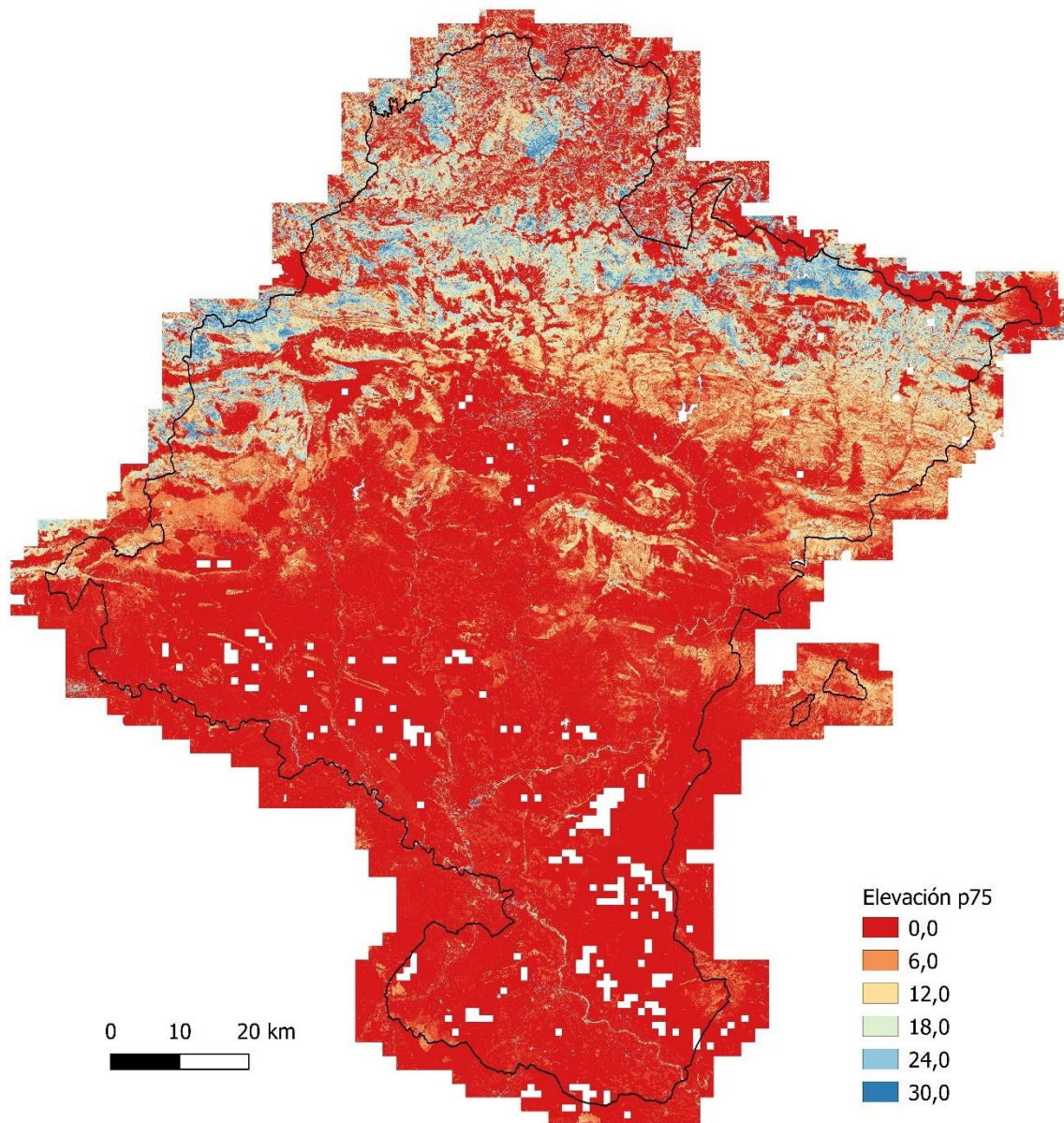


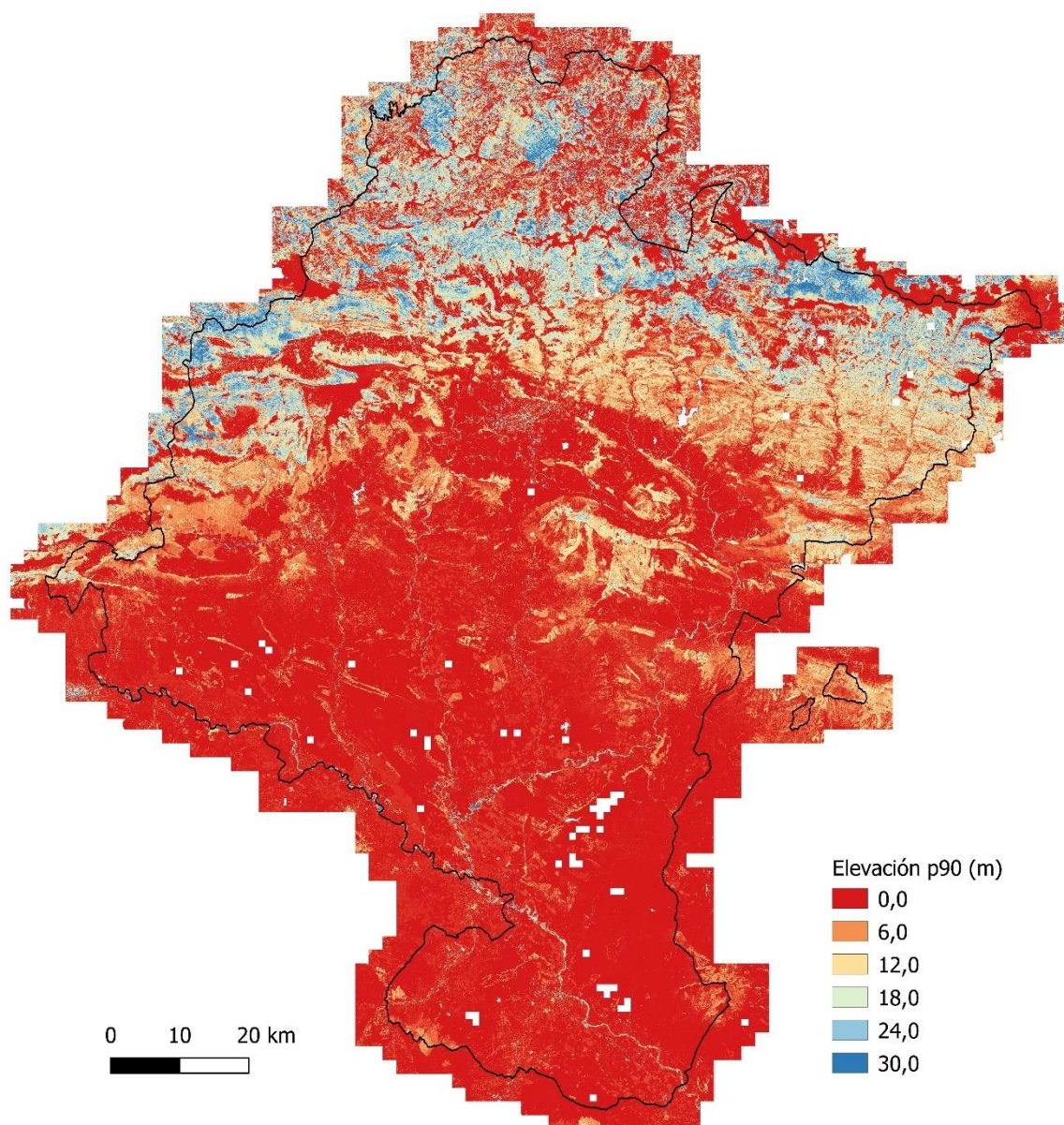




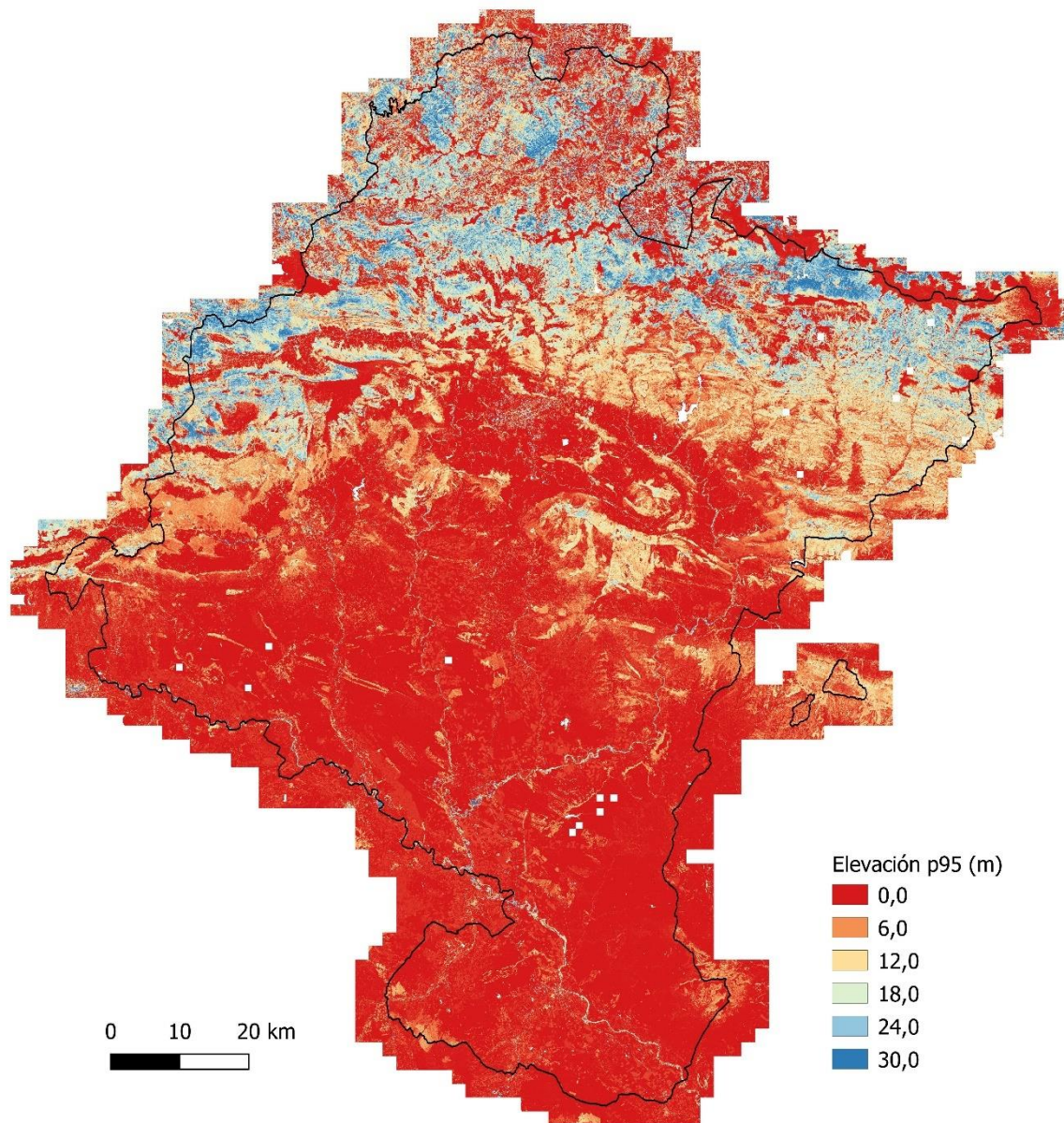


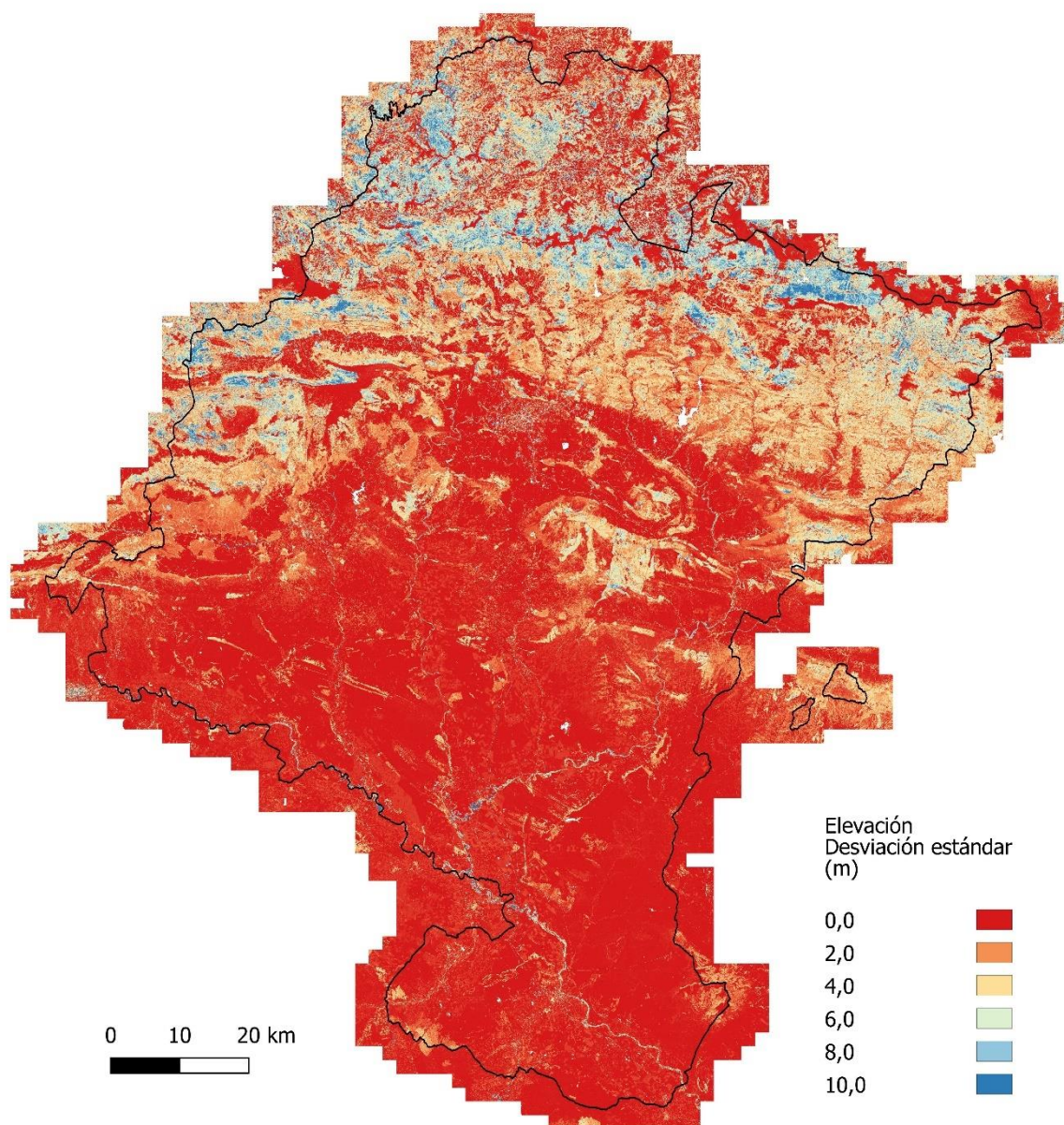




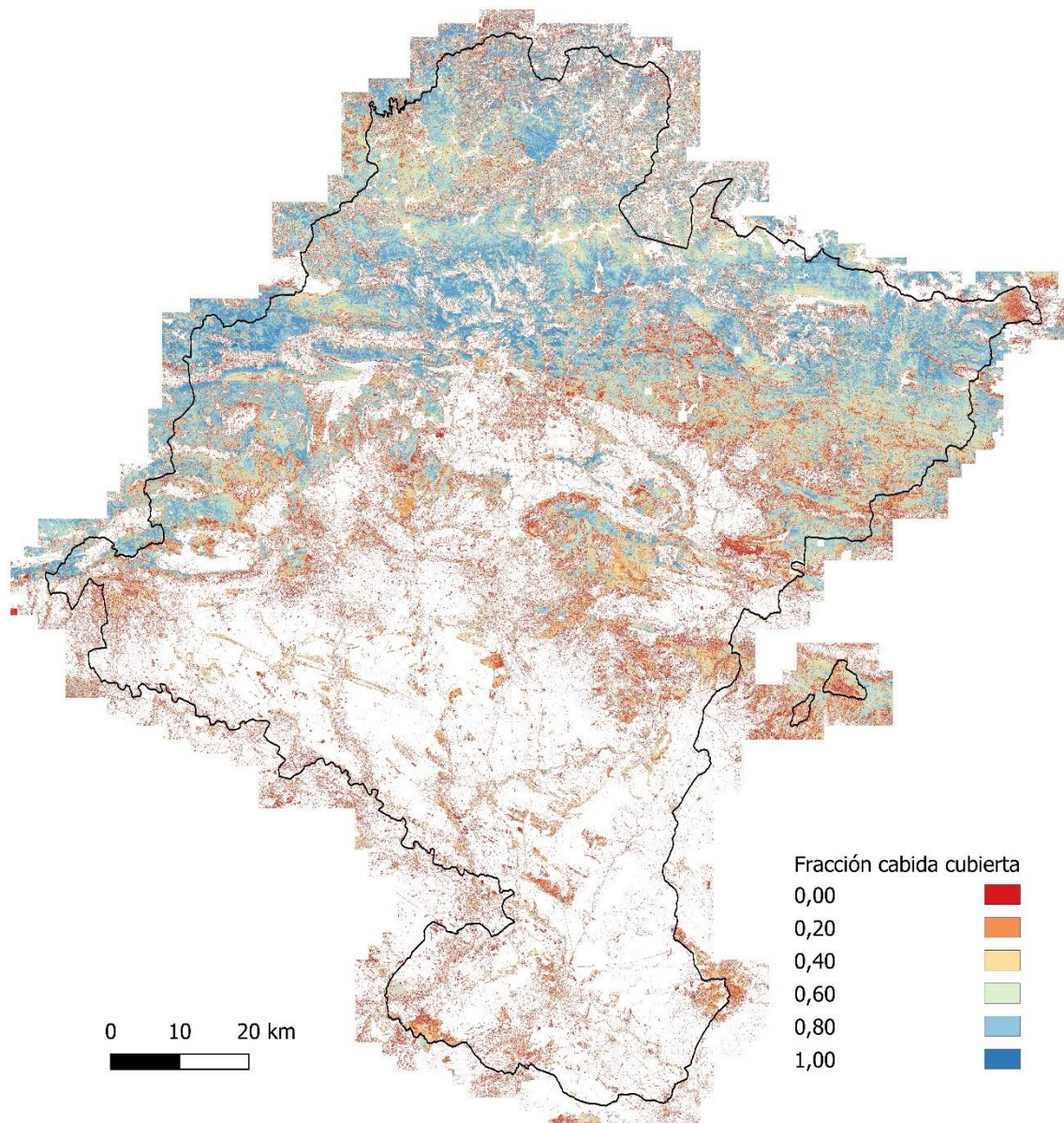


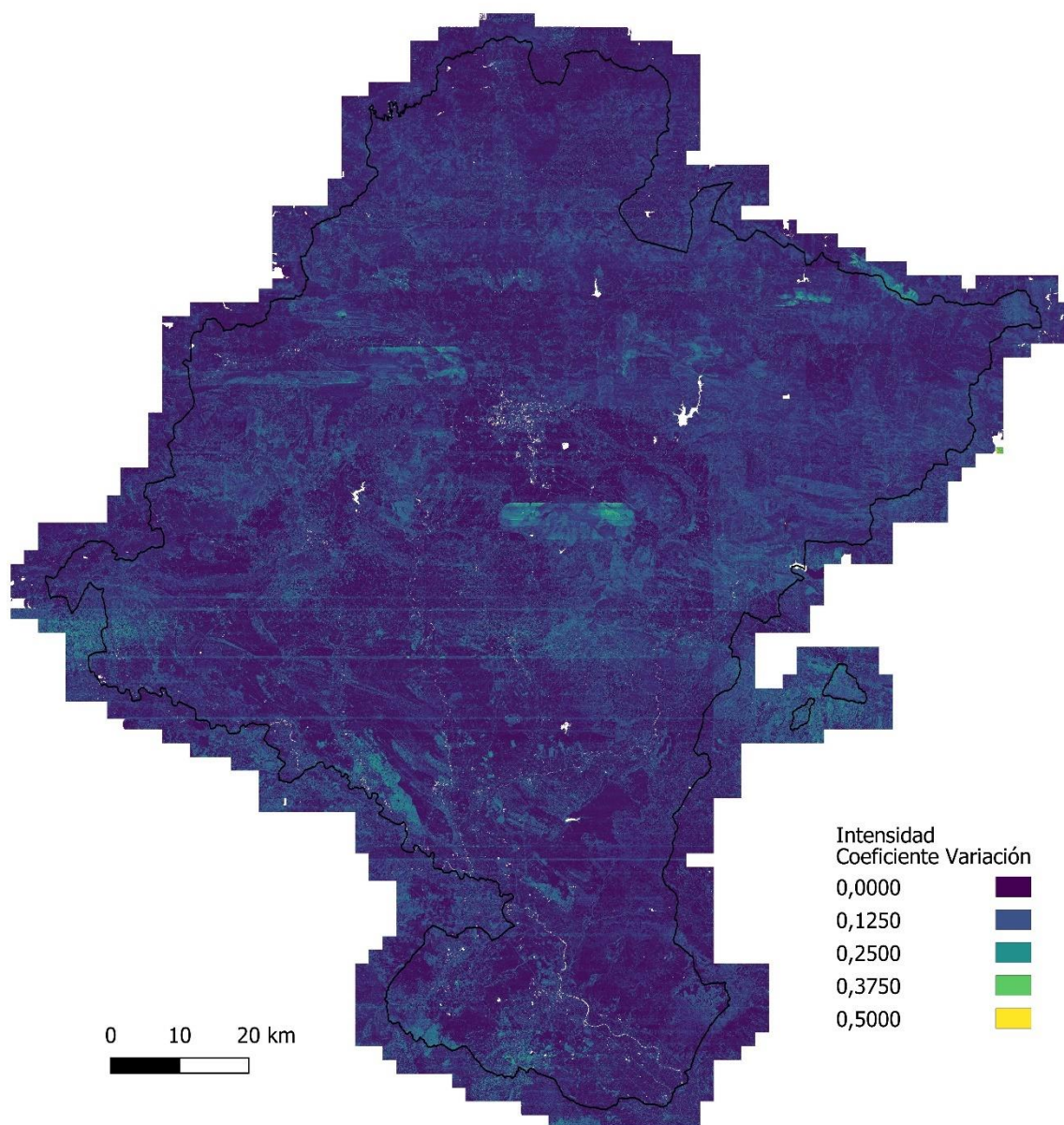




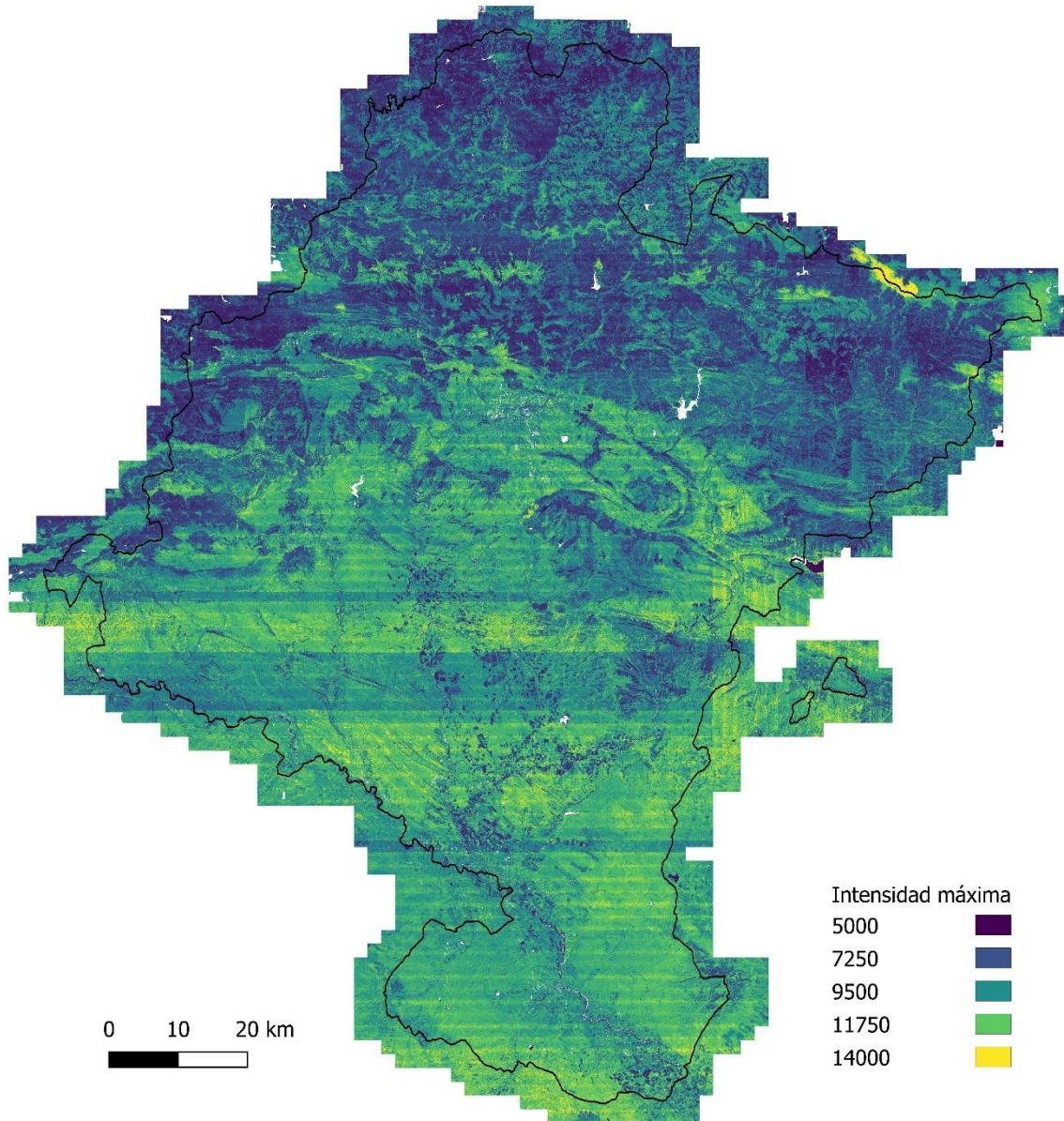


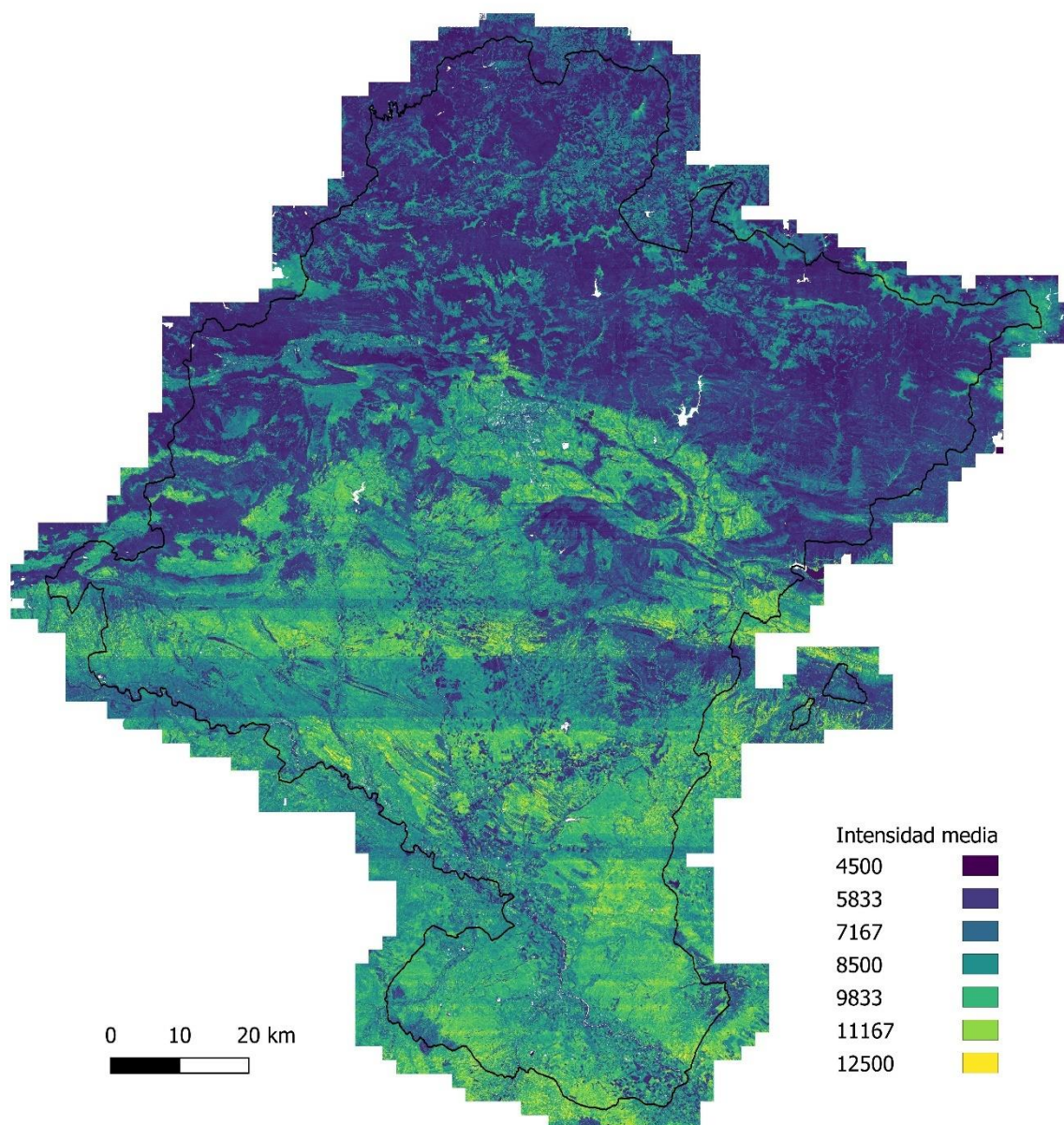




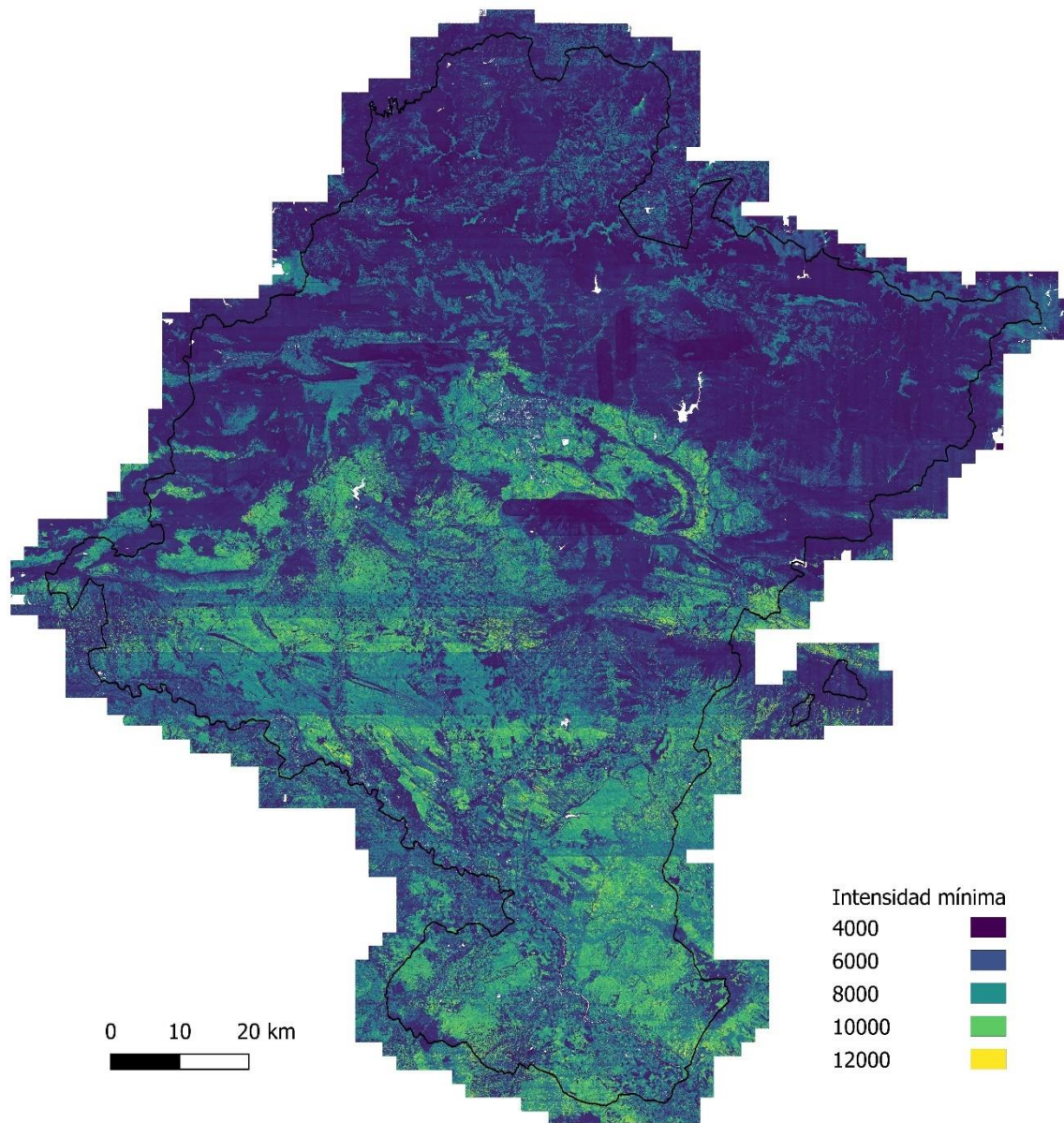


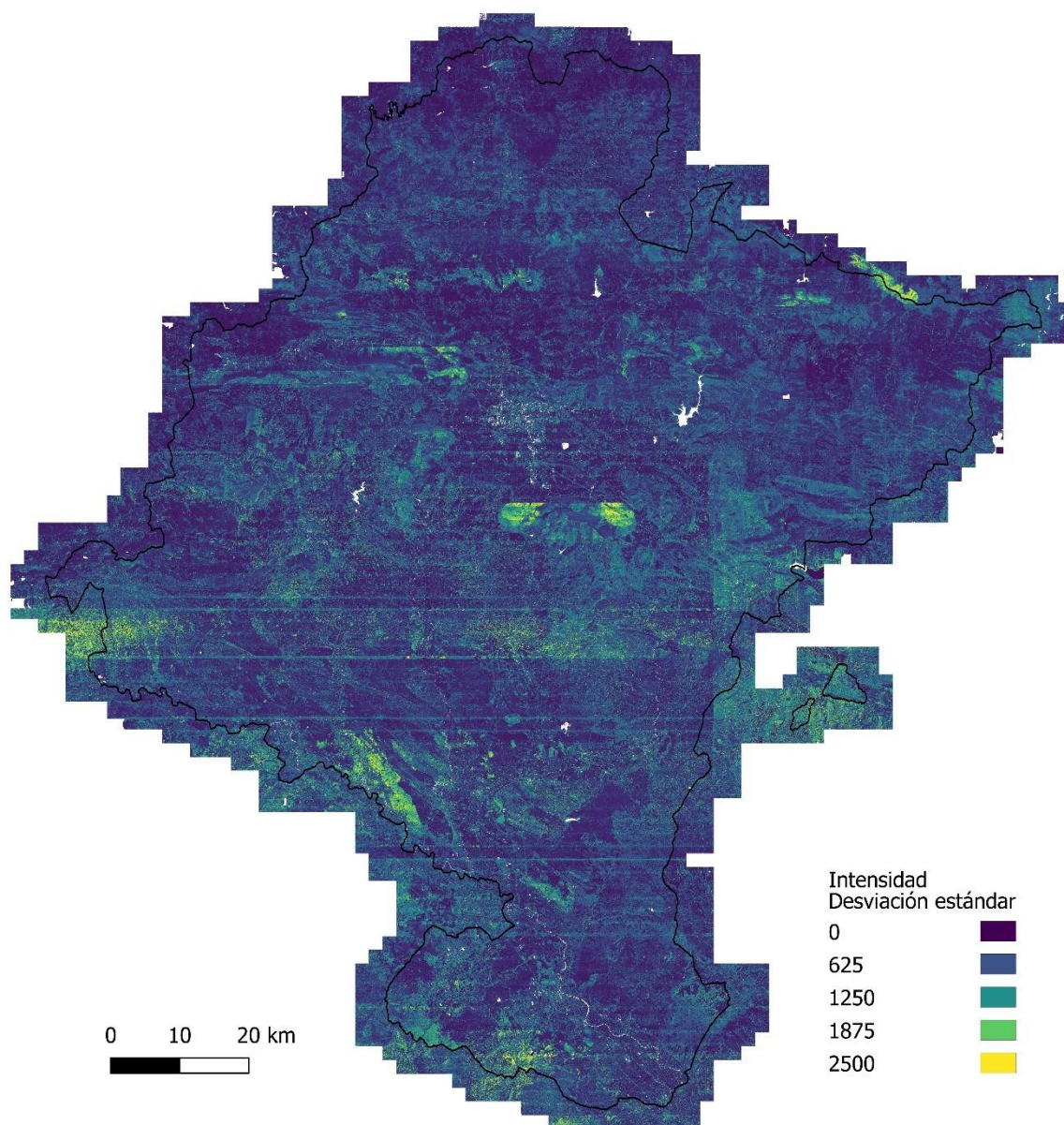




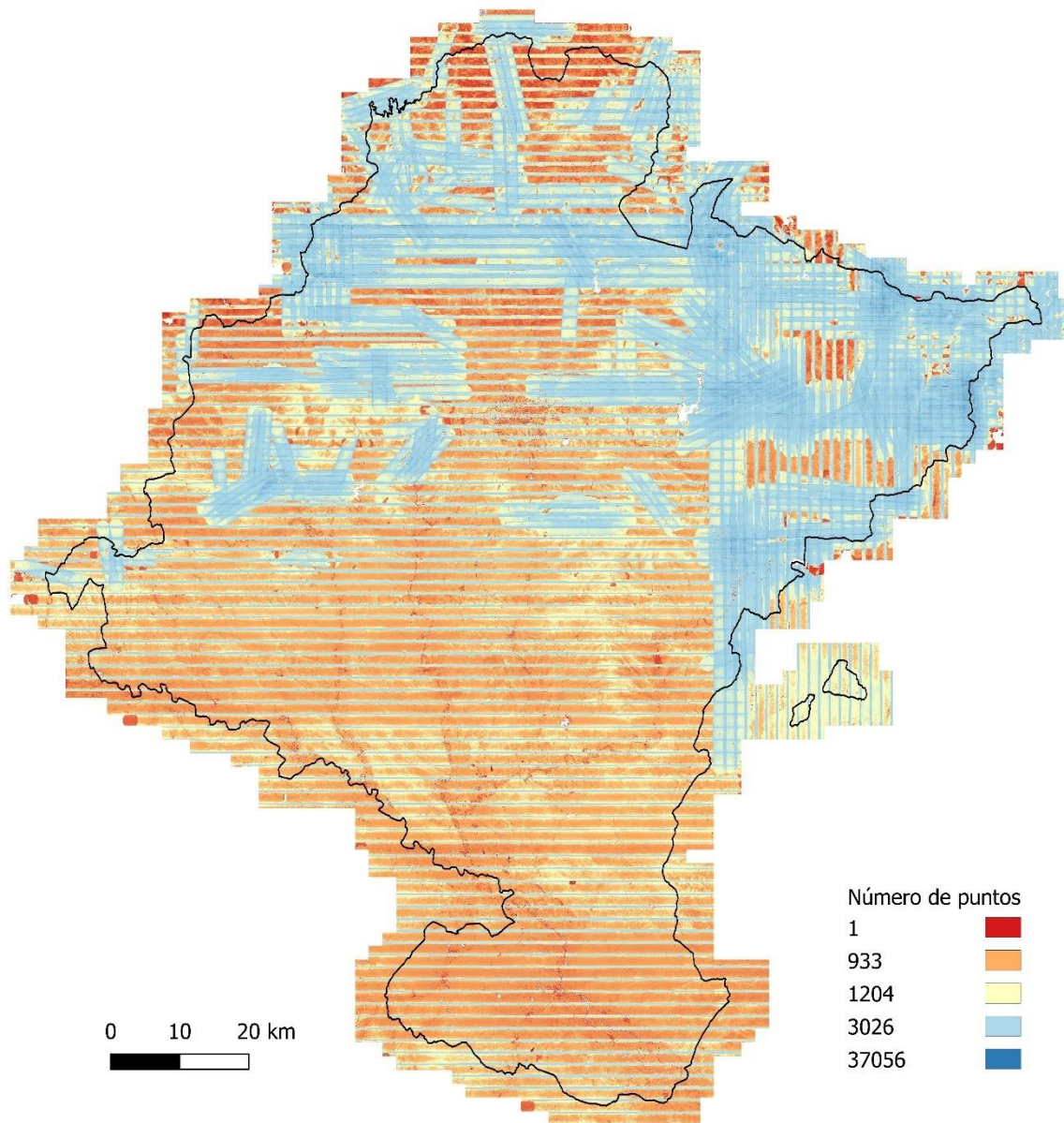


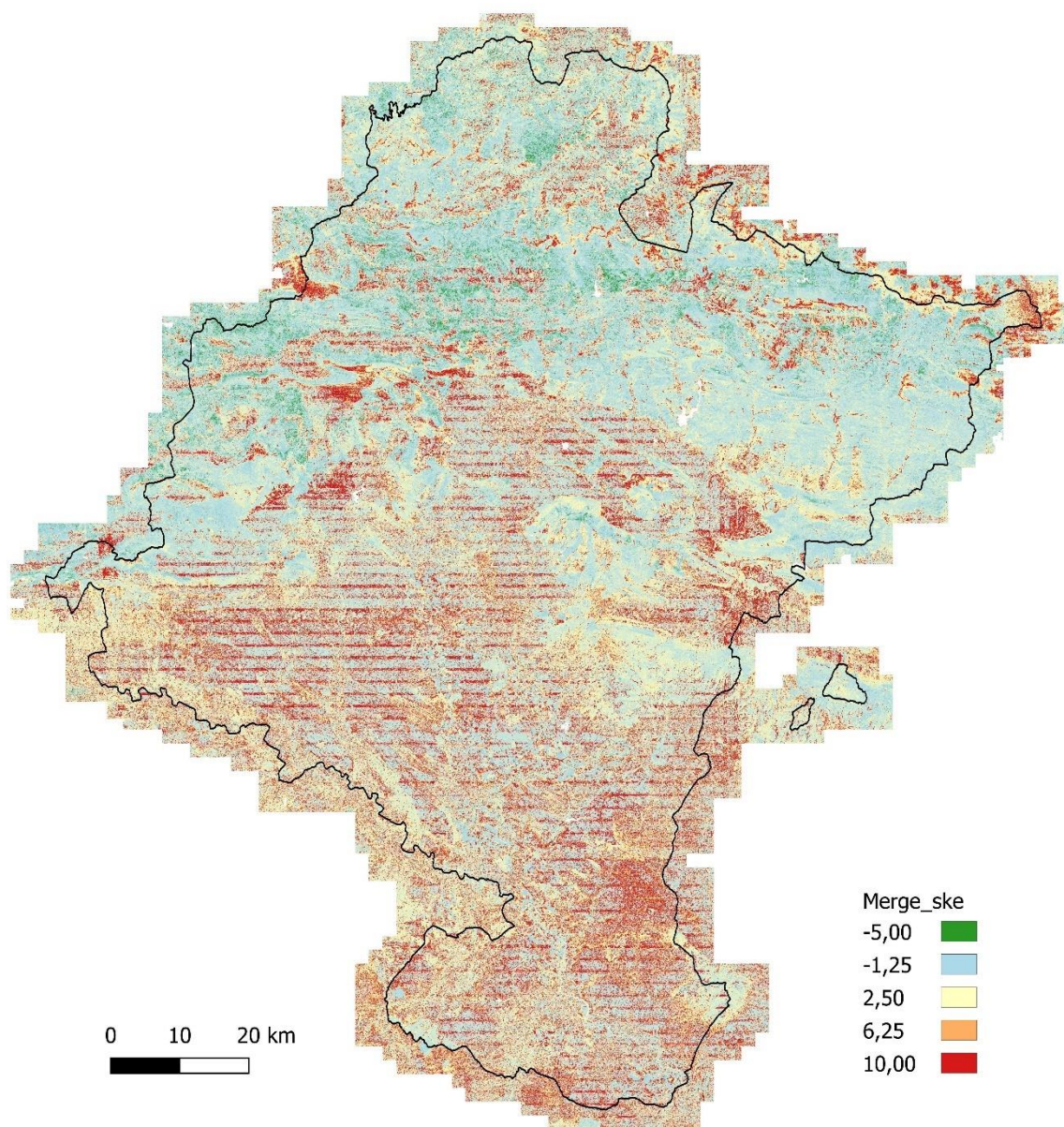












## X. ANEXO X:

### → Matrices de correlación de P.sylvestris

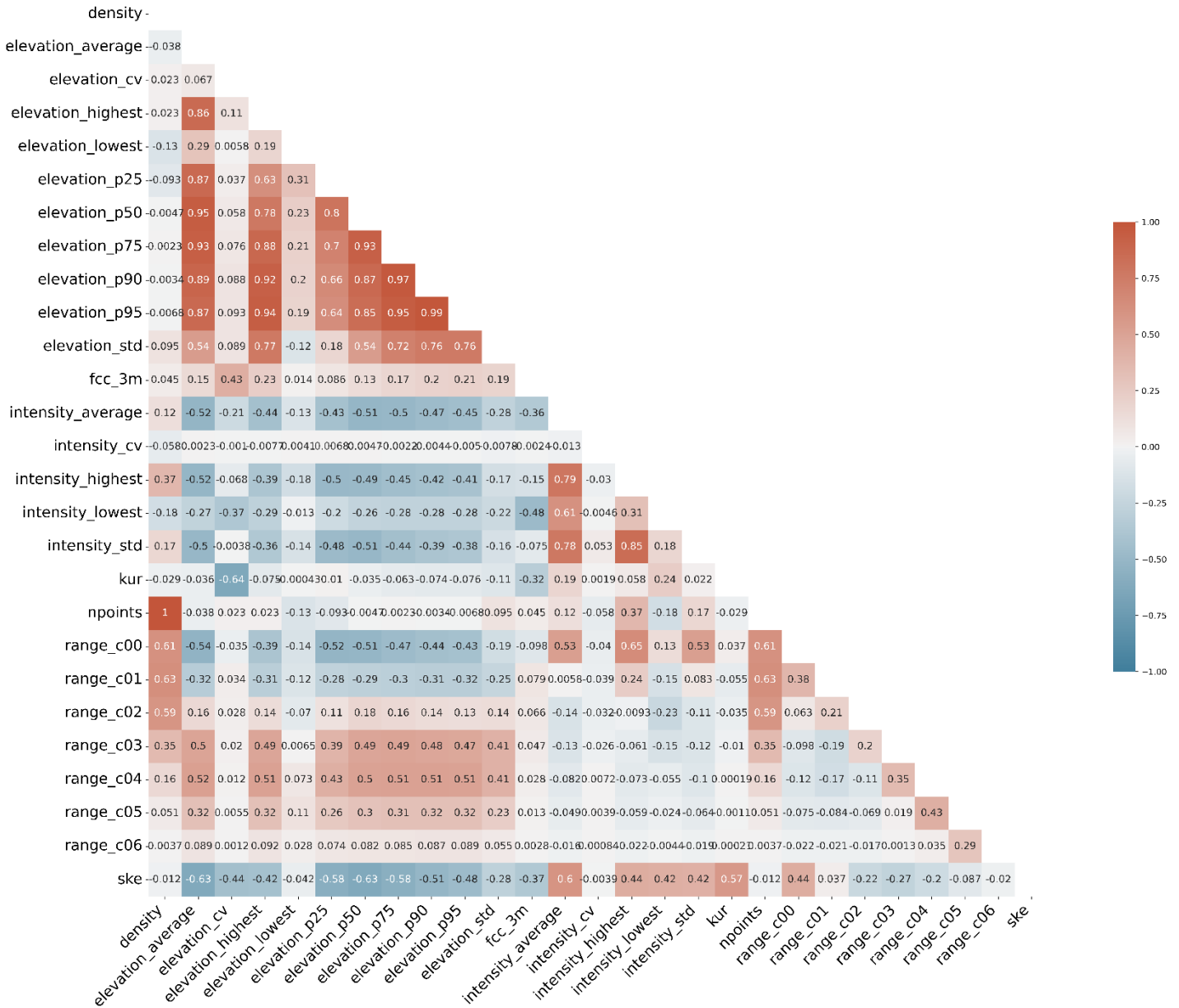


Figura 20: Correlaciones de las métricas LiDAR para P.sylvestris, método Pearson



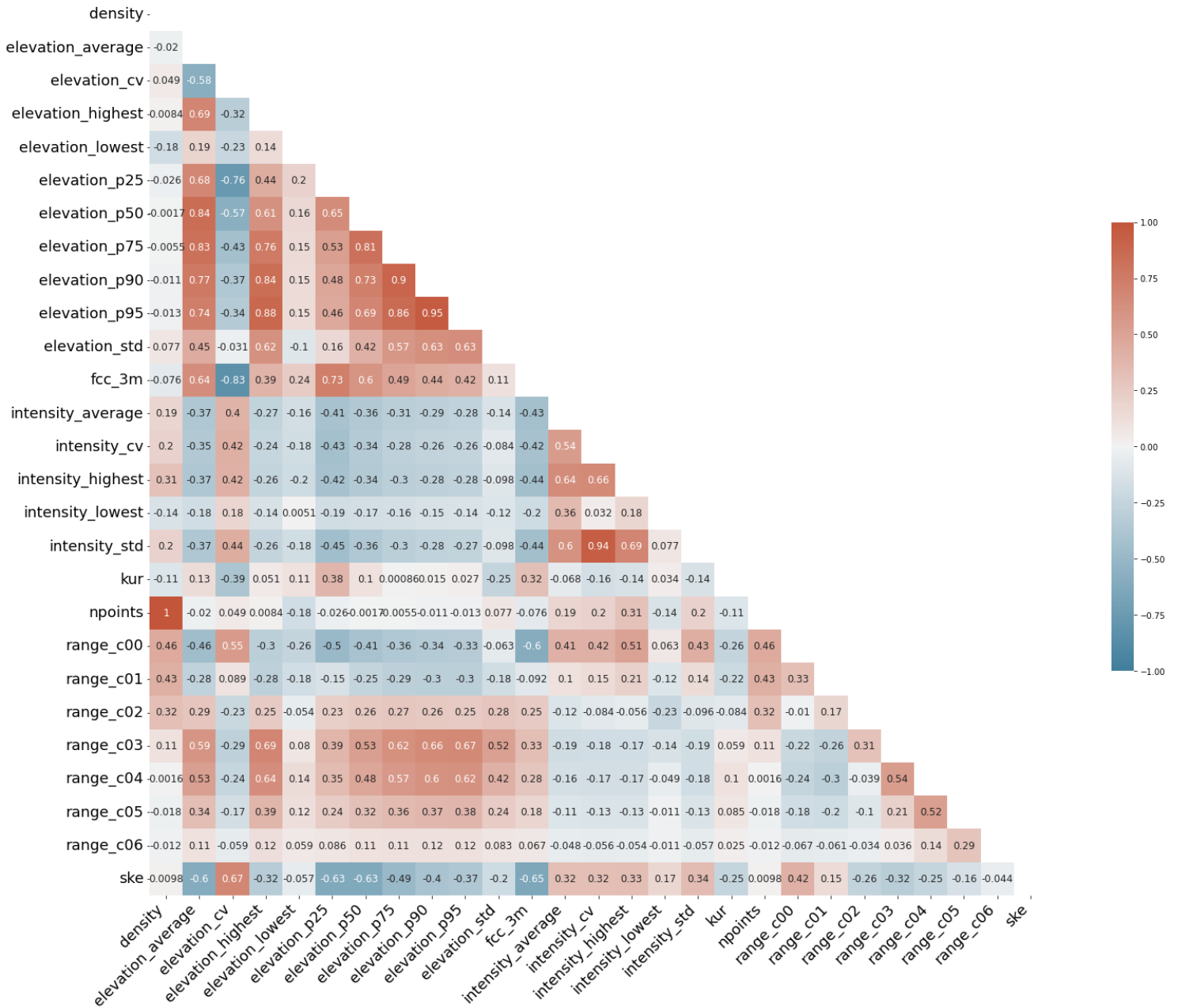


Figura 21: Correlaciones de las métricas LiDAR para *P.sylvestris*, método Kendall

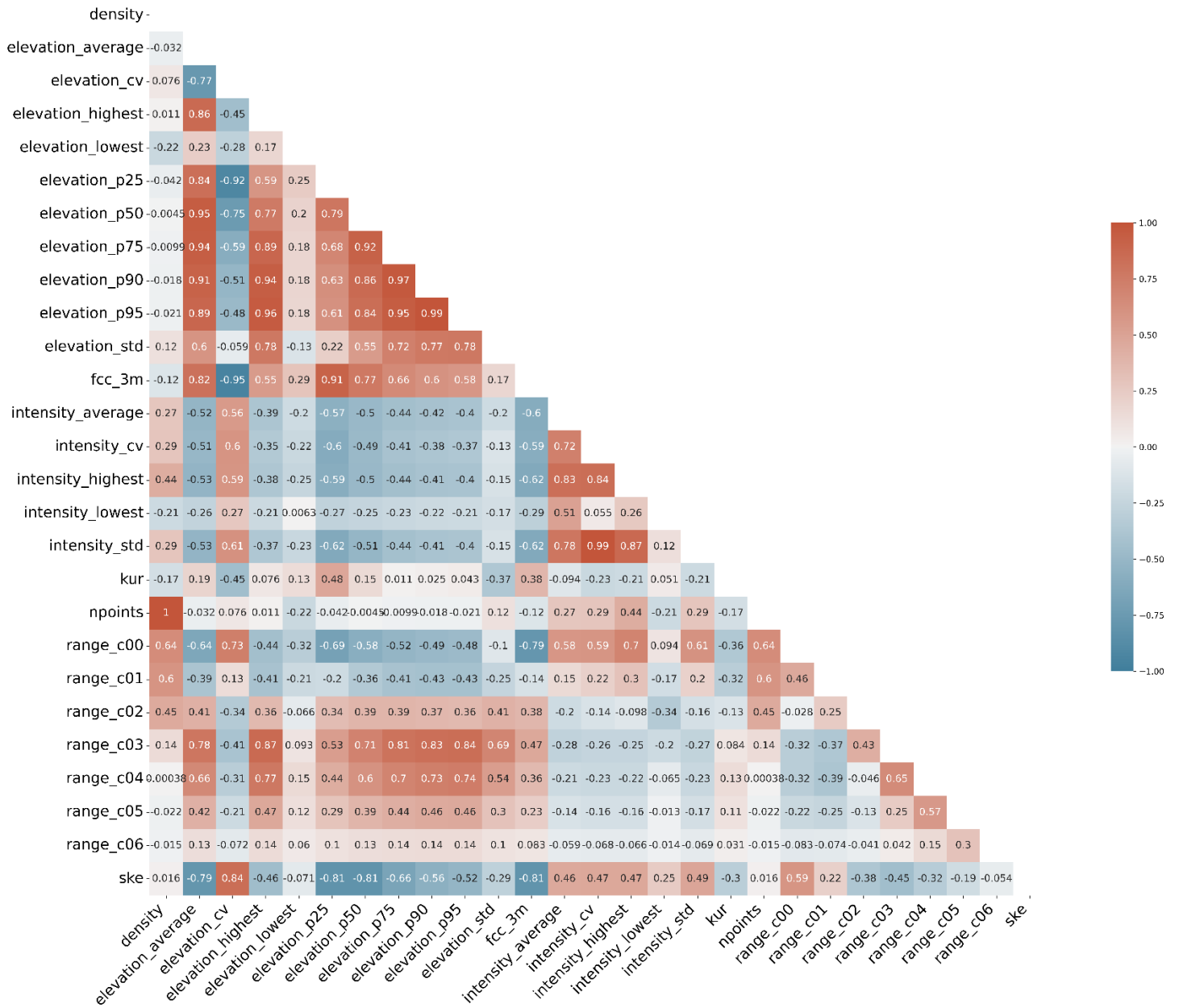


Figura 22: Correlaciones de las métricas LiDAR para *P.sylvestris*, método Spearman